PRISM: Prompt-Refined In-Context System Modeling for Financial Retrieval

Chun Chet Ng*✉
AI Lens
Kuala Lumpur, Malaysia
chunchet.ng@ailensgroup.com

Jia Yu Lim*
AI Lens
Kuala Lumpur, Malaysia

Wei Zeng Low
AI Lens
Kuala Lumpur, Malaysia

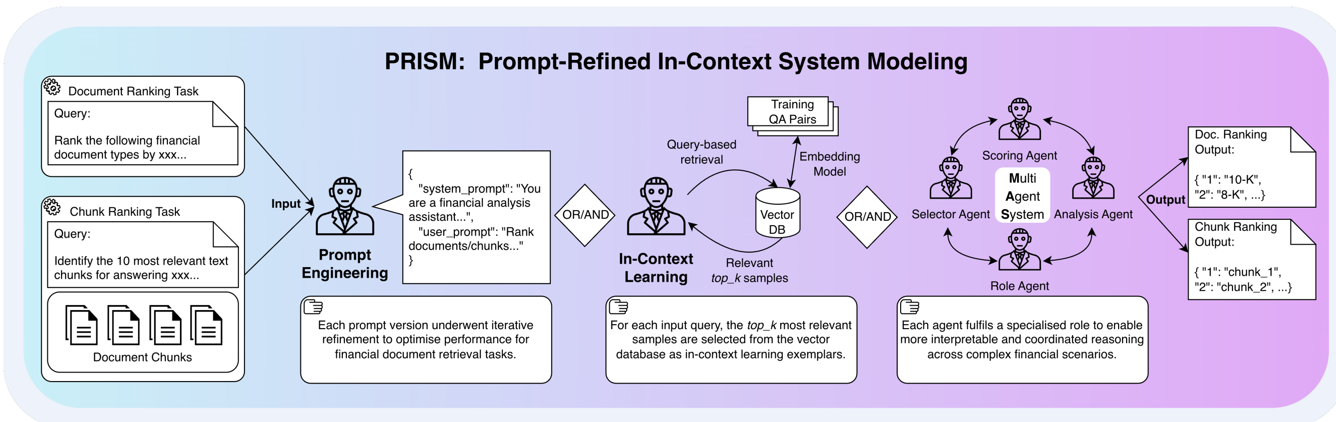


Figure 1: Overview of the PRISM framework. User queries are processed through three independent components: (1) prompt engineering encodes domain priors and reasoning scaffolds, (2) ICL retrieval augments prompts with semantically similar examples from the training set, and (3) optional multi-agent coordination decomposes ranking into specialized decision-making roles. The system outputs ranked documents and chunks for financial information retrieval.

Abstract

With the rapid progress of large language models (LLMs), financial information retrieval has become a critical industrial application. Extracting task-relevant information from lengthy financial filings is essential for both operational and analytical decision-making. We present PRISM, a training-free framework that integrates refined system prompting, in-context learning (ICL), and lightweight multi-agent coordination for document and chunk ranking tasks. Our primary contribution is a systematic empirical study of when each component provides value: prompt engineering delivers consistent performance with minimal overhead, ICL enhances reasoning for complex queries when applied selectively, and multi-agent systems show potential primarily with larger models and careful architectural design. Extensive ablation studies across FinAgentBench, FiQA-2018, and FinanceBench reveal that simpler configurations often outperform complex multi-agent pipelines, providing practical guidance for practitioners. On FinAgentBench, PRISM ranked third on the private leaderboard and achieved a best observed NDCG@5 of 0.71818 across repeated runs, while remaining the only training-free approach in the top three methods. We provide comprehensive feasibility analyses covering latency, token usage, and cost trade-offs to support deployment decisions. The source code and prompt templates are released at https://bit.ly/prism-ailens.

*Both authors contributed equally to this work.

Agentic AI Evaluation Workshop @ KDD 2026, Korea

CCS Concepts

• **Information systems → Information retrieval**.

Keywords

Financial Information Retrieval, Large Language Models, Prompt Engineering, In-Context Learning, Multi-Agent Systems

ACM Reference Format:
Chun Chet Ng, Jia Yu Lim, and Wei Zeng Low. 2026. PRISM: Prompt-Refined In-Context System Modeling for Financial Retrieval. In *Proceedings of The Workshop on Evaluation and Trustworthiness of Agentic AI at KDD 2026 (Agentic AI Evaluation Workshop @ KDD 2026)*. ACM, New York, NY, USA, 42 pages.

1 Introduction

Large language model (LLM)-based Information Retrieval (IR) has emerged as a transformative technology in the financial domain [33].

Extracting key information from lengthy financial documents remains labor-intensive and error-prone [32], driving research toward evidence-based retrieval where systems provide verifiable answers through Retrieval-Augmented Generation (RAG) workflows [7]. Despite these advances, challenges persist as financial documents are dense and domain-specific. Retrieval must span multiple granularities from full filings to individual passages, and no task-specific training signal is available in many practical settings. FinAgentBench was recently introduced to evaluate two tasks: document ranking to determine the relevance of financial documents, and chunk ranking to identify the most relevant segments within a document [8]. To address these challenges, we propose Prompt-Refined In-Context System Modeling (PRISM), a training-free framework that integrates precise system prompt engineering, in-context learning (ICL) augmentation, and multi-agent system modeling. Contributions of this work include:

(1) We present PRISM, a training-free framework that integrates prompt engineering, in-context learning, and multi-agent coordination for financial document and chunk ranking at multiple granularities, evaluated on FinAgentBench, FiQA-2018, and FinanceBench datasets.
(2) We conduct systematic ablation studies that examine how prompt design, ICL, and multi-agent coordination interact, identifying the conditions under which each component provides value and showing that simpler non-agentic configurations frequently outperform more complex multi-agent pipelines, particularly for fine-grained chunk ranking tasks.
(3) We provide latency, token usage, and cost analysis for training-free financial information retrieval, establishing reproducible baselines for accuracy-efficiency trade-offs.

2 Related Work

Prompt Engineering. Recent studies have shown that retrieval performance is highly sensitive to how an LLM is instructed [30]. The initial breakthrough in few-shot learning demonstrated that LLMs can perform tasks using in-context exemplars without gradient updates [4]. However, traditional few-shot approaches struggled with complex reasoning tasks. This limitation was addressed by Chain-of-Thought (CoT) prompting, which enhances reasoning by guiding LLMs through intermediate natural language reasoning steps [22]. The Reasoning and Acting (ReAct) framework further improves reliability by allowing LLMs to plan and execute reasoning steps iteratively, reducing hallucination [28]. Building on this, the Tree-of-Thoughts (ToT) framework enables LLMs to explore and evaluate multiple reasoning paths [27].

LLMs for Ranking. LLMs have also demonstrated strong potential as zero-shot rankers in IR, typically categorized into pointwise, pairwise, and listwise approaches [13]. Listwise methods, such as RankGPT, are often preferred for their balance between effectiveness and efficiency, but they face key challenges including limited input length and sensitivity to document order [20]. TourRank introduces a multi-stage grouping and tournament strategy to handle large candidate sets and improve robustness to input ordering [6]. Similarly, pairwise ranking compares candidate pairs directly, achieving linear complexity [17]. Well-instructed LLMs have demonstrated superior ranking performance compared to state-of-the-art supervised systems on major benchmarks such as TREC-DL and BEIR [20]. However, existing unsupervised methods remain constrained by LLMs' token limits when applied to long and dense financial texts [31].

In-Context Learning (ICL). In-context learning (ICL) enables LLMs to adapt to new tasks using input-output examples in the prompt, without gradient updates. Recent work views this mechanism as implicit Bayesian inference, where the model infers a latent concept that explains the observed examples [25]. Empirically, ICL accuracy scales with both the number and length of examples but is not strictly tied to correct input-label mappings. Randomly replacing labels only marginally affects performance [16], suggesting that ICL primarily benefits from contextual cues that define the label space, input distribution, and output format. The emergence of ICL has also been linked to properties of pre-training data such as temporal burstiness and diverse, low-frequency classes [5]. Meta-ICL aims to refine the model's implicit priors and adaptive strategies by presenting multiple tasks sequentially [9]. Embedding-based ICL retrieval for financial IR with commercial LLMs remains relatively underexplored compared to IR tasks in the general domain.

Agentic Information Retrieval. Agentic information retrieval represents a new paradigm where LLM agents dynamically manage information access through iterative cycles of observation, reasoning, and action, distinguishing it from traditional IR architectures [29]. It is commonly implemented using multi-agent systems (MAS), which coordinate specialized agents for reflection, planning, and tool use to achieve collective intelligence and outperform single-agent approaches [19]. LLM agents have also been applied to ranking tasks, demonstrating that retrieval and ranking agents can effectively coexist [26]. Despite its potential, Agentic IR faces challenges in coordinating complex MAS interactions, reducing the high inference cost of LLMs, and evaluating dynamic agentic behaviors in financial retrieval.

3 Datasets

We evaluate PRISM and perform Exploratory Data Analysis (EDA) on three representative financial datasets covering complementary task characteristics: ranking granularities (document vs. chunk vs. passage), task types (ranking vs. question answering), and reasoning demands (lexical matching vs. numerical analysis).

FiQA-2018 [15] is a financial opinion QA dataset from the BEIR benchmark [21], containing questions sourced from financial microblogs, news headlines, and reports. The task requires passage reranking to identify relevant answers. NDCG@10 and Recall@100 are used as evaluation metrics, compared against baselines including BM25 and cross-encoder rerankers.

FinanceBench [14] is a question-answering benchmark requiring numerical reasoning over financial documents. Unlike ranking tasks, it tests whether systems can extract correct factual answers from SEC filings. We evaluate under oracle conditions (relevant passages provided) to isolate the reasoning component from retrieval.

FinAgentBench [8] is a large-scale dataset for financial IR comprising 2023–2024 SEC filings from the EDGAR database. It supports two tasks: (i) ranking five document types (10-K, 10-Q, 8-K, DEF 14A, Earnings Transcripts) by relevance to a query, and (ii) ranking text chunks within documents. The dataset includes 4,986 document-ranking and 18,855 chunk-ranking training samples, with 200 validation samples per task.

3.1 EDA on FinAgentBench

Table 1: Document distribution across ranks.

Doc. Type	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
10-K	**2402**	1546	507	207	324
Earnings	**1360**	726	948	893	1059
10-Q	554	**1558**	1427	866	581
8-K	97	407	1091	**1808**	1583
DEF 14A	573	749	1013	1212	**1439**

Document Ranking Data Analysis. We first conduct a systematic evaluation across all document types in the document-ranking split of FinAgentBench. Table 1 summarizes the distribution of document types across ranking positions. Documents such as 10-K and 8-K appear more frequently near the top and bottom of the rankings, respectively, suggesting that information density and relevance vary across document types. Table 2 shows the six most frequent keywords found in Rank 1 documents for each type, revealing distinct lexical signatures across filing categories. These differences in vocabulary concentration across document types highlight the need for adaptive retrieval strategies, which inspired PRISM. Further details are provided in Appendix A.1.1.
Chunk Ranking Data Analysis. We conducted a frequency analysis to better understand chunk-level characteristics in the dataset. As shown in Figure 2, relevant chunks are generally more information dense than irrelevant ones. The wider interquartile range (IQR) and higher maximum values, along with the long-tail distribution, suggest that adaptive retrieval methods are needed to handle substantial variability in chunk length. The relatively small proportion of relevant chunks highlights the fine-grained and selective nature of financial retrieval. Our EDA shows that relevant answers are concentrated in specific document types and in a minority of information-dense chunks, requiring models to reason over and manage large variations in chunk length.

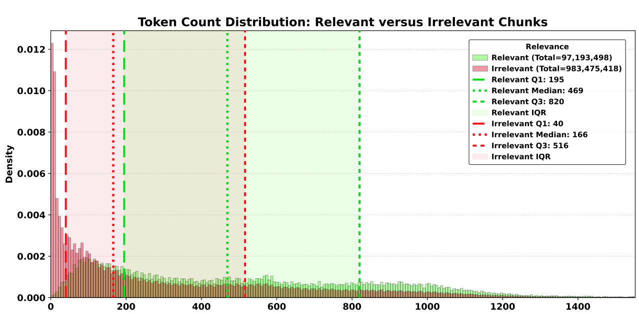


Figure 2: Token count distribution of chunks.

3.2 EDA on FiQA-2018

FiQA-2018 contains 57,638 documents and 648 queries with 1,706 relevance judgments. With only 2.96% of the corpus labeled as relevant and a sparsity of 99.99%, FiQA-2018 exhibits extreme label sparsity characteristic of real-world retrieval tasks. Most queries have fewer than three relevant documents, with average query and document lengths of 11 and 133 words, respectively. This characteristic places strong emphasis on semantic understanding over lexical overlap, disadvantaging keyword-based models such as BM25 when queries and documents share limited surface-form vocabulary. Dense retrieval models are better suited to this task because they capture latent semantic relevance beyond exact term matching (see Appendix A.2 for detailed corpus statistics).

3.3 EDA on FinanceBench

FinanceBench contains 150 documents across four types: 10-K (112), 10-Q (15), 8-K (9), and Earnings Transcripts (14), spanning 32 unique companies. These document types exhibit distinct lexical signatures and clearly different document structure: 10-K reports concentrate heavily on company- and industry-specific vocabulary, while Earnings Transcripts show the most diverse vocabulary, reflecting broader analyst Q&A session patterns. This intra-corpus diversity means that a single retrieval strategy optimized for one document type may systematically underperform on others, underscoring the need for type-aware retrieval approaches. These differences informed the construction of our type-specific agent prompts (see Appendix A.3 for the top keywords by document type and supporting lexical analysis).

4 PRISM

Guided by the EDA findings above, PRISM integrates three components whose interactions are illustrated in Figure 1: (1) prompt engineering that encodes domain priors and reasoning scaffolds, (2) ICL retrieval that dynamically augments prompts with semantically relevant examples, and (3) multi-agent coordination that decomposes ranking into specialized decision-making roles.

4.1 Prompt Engineering

Prompt engineering in PRISM serves as an integrated system module that governs how LLMs process ranking tasks. This module encodes domain-specific priors derived from our EDA, such as document-type distributions and keyword patterns, into structured instructions that guide the model's reasoning under long-context retrieval constraints. We introduce logical scaffolding inspired by ReAct, CoT, and ToT to promote explicit reasoning before producing final answers. Four prompt variants, P_1–P_4, were systematically designed to assess the impact of incremental modifications and progressively enhance reasoning structure. The progression from P_1 to P_4 reveals that structural reasoning refinements consistently outperform content-based augmentations: P_4 achieves the best overall results across both ranking tasks, confirming that reasoning clarity outweighs informational density. For document ranking, the progression moves from domain-guided heuristics to combined reasoning frameworks; for chunk ranking, prompts evolve from a pure ReAct framework to integrated multi-strategy approaches. Specifically, the prompt iterations are as follows:

Table 2: Top six keywords appearing in queries for which each document type was ranked first in FinAgentBench. The ratio indicates the frequency of the keyword relative to the entire corpus.

DEF 14A		10-K		10-Q		8-K		Earnings	
Keyword	Ratio	Keyword	Ratio	Keyword	Ratio	Keyword	Ratio	Keyword	Ratio
dependency	88.40%	evolved	40.98%	quarter	20.00%	compensation	95.45%	asked	87.68%
concentration	88.24%	recurring	40.57%	recurring	6.29%	burn	41.74%	questions	86.56%
exist	87.73%	ratio	39.40%	evolved	6.12%	award	41.74%	metrics	78.26%
risks	80.51%	time	39.22%	time	5.99%	manage	41.54%	customer	74.62%
market	60.77%	reporting	38.19%	ratio	5.97%	availability	40.60%	guidance	52.97%
share	57.12%	period	36.01%	revenue	4.02%	share	27.21%	offered	52.80%

- P_1: For document ranking, introduces domain-specific keyword hints and statistical cues to guide relevance assessment. For chunk ranking, employs a ReAct-style framework that interleaves internal reasoning with action steps, requiring the model to think step-by-step before outputting ranked indices.
- P_2: For document ranking, enhances P_1 with detailed statistical keyword associations (e.g., word frequency ratios per document type) to improve discriminative power. For chunk ranking, combines Chain-of-Thought (CoT), ReAct, and Tree-of-Thought (ToT) strategies, introducing multi-expert simulation where three analysts reason collaboratively and prune incorrect paths.
- P_3: Adopts a unified multi-strategy framework for both ranking tasks, integrating CoT decomposition, ReAct alternation, and ToT multi-expert simulation. Emphasizes explicit step-by-step reasoning with domain-specific hints for financial concepts.
- P_4: Refines the P_3 prompt with streamlined instructions and consistent terminology across both ranking tasks, maintaining the combined CoT, ReAct, and ToT reasoning strategies.

Full templates are provided in Appendix A.14 and Appendix A.4.

4.2 In-Context Learning Sample Retrieval

The second module enhances the system with few-shot learning through a simplified RAG pipeline. As our EDA reveals uneven lexical distributions and varying information density across document types and chunks (Tables 1–2), incorporating dynamically retrieved, well-chosen examples that reflect these domain priors gives LLMs a stable in-context reference point, helping them identify relevant documents earlier, reason more consistently, and reduce hallucinations in long-context settings.

Exemplar Store Construction. We first construct an offline exemplar store from the training dataset. Each training sample consists of a query paired with documents or chunks and their ground-truth relevance rankings. We embed all query-document/chunk pairs using OpenAI's text-embedding-3 model and index them in a FAISS vector store. This store remains fixed and serves as the source for retrieving few-shot examples.

Inference-Time Retrieval. During inference, each new query is embedded using the same model and the top-k most semantically similar exemplars are retrieved from the FAISS store. These exemplars, including example queries and their ground-truth rankings, are formatted as few-shot demonstrations and prepended to the system prompt, adapting the prompt to the query. The exemplar store is kept separate from the document and chunk corpus being ranked, ensuring retrieved examples serve purely as reasoning scaffolds without contaminating the ranking candidates. Empirically, ICL performs best when retrieved exemplars are semantically close to the target query, as closer examples provide more relevant reasoning patterns for the model to follow. Full retrieval implementation details and shot-count sensitivity are provided in Appendix A.5.

4.3 Multi-Agent System (MAS) Modeling

The third module introduces a structured MAS to decompose reasoning into specialized agent roles, aiming to reduce cognitive load, improve calibration, and mitigate hallucinations through role specialization and graph-level coordination. We include MAS to systematically study the conditions under which multi-agent coordination provides value, specifically for larger models and document-level tasks, and when coordination overhead becomes a limiting factor for smaller models and chunk-level tasks. A key contribution of our work is the empirical characterization of MAS.

Architectural Variants. For chunk ranking, the MAS defines four progressively simplified architectures (A_1–A_4). A_1 is a direct scoring ensemble with four parallel agents (CEO, Financial Analyst, Operations Manager, Risk Analyst) that independently evaluate candidate chunks before reaching consensus through score averaging. A_2 is a three-stage pipeline with Noise Remover → Candidate Selector → scoring ensemble (Relevance Scorer, Contextual Reasoner, Evidence Extractor, Diversity Agent). This deeper architecture tends to suffer from over-filtering and cascading errors. A_3 is a two-stage pipeline with Quick Filter → three scoring agents (Relevance Scorer, Contextual Reasoner, Evidence Extractor), balancing depth and stability. A_4 is a minimal configuration with two scoring agents (Financial Analyst, Risk Analyst), reducing interaction complexity and error propagation. Further implementation details for each variant are included in Appendix A.6.

5 Experiments

We evaluated four GPT models (GPT-4o-mini, GPT-4.1, GPT-5-mini, and GPT-5) and four open-source models (DeepSeek-V4-Flash [11], gpt-oss-120b [1], grok-4-20-reasoning [24], and Llama-4-Maverick-17B-128E-Instruct-FP8 [2]), using OpenAI's text-embedding-3-small (TE3-S) and text-embedding-3-large (TE3-L) for ICL retrieval. Our evaluation spans three datasets chosen to cover complementary task characteristics: FinAgentBench (document and chunk ranking with SEC filings), FiQA-2018 (passage reranking for financial QA), and FinanceBench (factual QA requiring numerical reasoning). The training-free nature of PRISM nevertheless allows straightforward adaptation to other LLM providers and deployment settings. Full implementation details are provided in Appendix A.7.

5.1 Quantitative Results

Table 3: FinAgentBench leaderboard results.

Rank	Team	NDCG@5 Score (Private)	Code Released	Fine-tuning
1	memex	0.72497	N/A	Yes
2	Yuzhen Hu	0.72328	N/A	Yes
3	Ours	0.71181†	Yes	No
4	rlawnsxo	0.70601	N/A	N/A
5	currio	0.70195	N/A	N/A
6	kaggleslasher	0.69985	N/A	N/A
7	Gautam Jajoo	0.69774	N/A	N/A
8	RagTag	0.69173	N/A	N/A
9	ART YOUNG	0.69162	N/A	N/A
10	liangjiang chen	0.69081	N/A	N/A

† Score from the submission selected during the competition. Our best observed score across repeated runs of the same configuration (Run 19) is 0.71818 in Table 6.

FinAgentBench. Performance is measured using NDCG@5, with results reported on the Kaggle test set (30% public and 70% private subsets). As ground-truth labels for the public and private test sets were not disclosed, we report the combined document-and-chunk NDCG@5 scores as computed by the Kaggle evaluation system, separately for each subset. Following the competition's conclusion, this evaluation system became unavailable, preventing further experiments. Table 3 presents the top-10 leaderboard on the private subset. On the private subset, PRISM ranked third overall while being the only training-free method in the top three. The top two methods relied on extensive fine-tuning: the first-ranked memex team deployed a pool-of-experts of five LLMs for pre-filtering and 65 agents for chunk retrieval and reranking, with an embedding model fine-tuned for 30 epochs on FinAgentBench training data; the second-ranked team used three MiniLM models fine-tuned separately for document retrieval, chunk retrieval, and cross-encoder reranking. In contrast, PRISM achieves competitive performance using only off-the-shelf LLMs with prompt engineering and ICL, with no task-specific training. Our selected leaderboard submission scored 0.71181, and repeated runs of the same configuration (Run 19) reached a best score of 0.71818, trailing the top fine-tuned method by only $\Delta = 0.007$ NDCG@5 while leading the remaining teams by approximately 0.02 on average, indicating that training-free approaches can be competitive with fine-tuned systems.

FiQA-2018. PRISM achieves substantial improvements over existing methods on the test set as shown in Table 4, with relative gains of 78.59% in NDCG@10 and 52.24% in Recall@100 compared to the best-performing reranking baseline, BM25+Cross-Encoder (CE). Baselines from the BEIR benchmark [21] are reported directly from published results under the BEIR evaluation protocol. With GPT-5, PRISM consistently achieves NDCG@10 above 0.60 and Recall@100 near 0.82 across all prompt variations. The impact of ICL varies significantly by model capability, revealing important insights for practical deployment. ICL proves particularly beneficial for models with more limited reasoning capabilities, as evidenced by GPT-4.1's consistent performance gains across all prompts (e.g., 23.5% improvement in NDCG@10 for the P_1 configuration). This enhancement occurs because ICL provides concrete demonstrations of the task structure and expected reasoning patterns, effectively compensating for limited inherent reasoning capacity by offering explicit templates to follow. In contrast, GPT-5-mini and GPT-5 exhibit stable performance with marginal variation when ICL is applied, indicating their robust capacity to generalize from task instructions alone without requiring extensive demonstration examples except on highly specialized tasks. This differential ICL sensitivity suggests that smaller or less capable models can be substantially enhanced through careful example selection, while more advanced models already possess sufficient task understanding capabilities. The open-source model evaluations using P_4 further validate PRISM's generalizability beyond proprietary systems. All OSS models substantially outperform the BM25+CE baseline under both zero-shot and ICL-5 conditions, with NDCG@10 ranging from 0.552 to 0.583, trailing GPT-5 by a modest margin. Notably, gpt-oss-120b achieves the strongest OSS result of 0.5833 NDCG@10 under ICL-5, approaching the performance of GPT-5-mini. The effect of ICL across OSS models is consistent but marginal, with ICL-5 yielding small improvements over zero-shot across all four models, suggesting that these models already possess sufficient instruction-following capacity to perform the reranking task without extensive demonstration examples. The relatively uniform performance across architecturally diverse OSS models, spanning mixture-of-experts, dense, and reasoning-specialized designs indicates that PRISM's prompt-driven approach is robust to underlying model architecture, reinforcing its practical accessibility for deployments outside proprietary cloud ecosystems. Comprehensive model comparisons against BEIR baselines and feasibility analysis are provided in Appendix A.8.2.

Table 4: FiQA evaluation results.

Prompt	Model	ICL	NDCG@10	Recall@100
N/A	BM25+CE (Baseline)	N/A	0.3470	0.5390
Closed-Source Models				
P_1	GPT-5	ICL-5	0.6104	0.8083
P_2	GPT-5	ICL-5	0.6027	0.7941
P_3	GPT-5	ICL-5	0.6197	0.8206
P_4	GPT-5	ICL-5	0.6097	0.8021
Open-Source Models				
P_4	DeepSeek-V4-Flash	ICL-5	0.5778	0.7905
P_4	gpt-oss-120b	ICL-5	0.5833	0.6758
P_4	grok-4-20-reasoning	ICL-5	0.5741	0.7881
P_4	Llama-4-Maverick-17B-128E-Instruct-FP8	ICL-5	0.5581	0.7726

FinanceBench. PRISM is evaluated under oracle context conditions as shown in Table 5, using a reimplementation of the strongest baseline reported in the original paper with a comparable model family. PRISM achieved 98% accuracy (147/150) with the P_1 prompt, surpassing the GPT-5 oracle baseline of 94% (141/150). While prompt variants performed similarly on reranking datasets, QA favored the simpler P_1 formulation, indicating that optimal prompting is task-dependent. The limited utility of ICL in FinanceBench can be attributed to its task-specific characteristics, in contrast to its substantial benefits in reranking tasks. FinanceBench queries require finding specific answers within individual documents rather than comparing multiple candidates, making similarity-based ICL selection less effective at retrieving truly relevant question-answer exemplars. This results in ICL examples that may misdirect rather than guide the model, explaining why non-ICL configurations often perform comparably or better. All GPT-5 configurations exceeded 90% accuracy. Among open-source models, gpt-oss-120b and grok-4-20-reasoning also exceeded 95%, while DeepSeek-V4-Flash reached 83.3% and Llama-4-Maverick-17B-128E-Instruct-FP8 lagged noticeably, suggesting that smaller instruction-tuned models may need further adaptation for structured financial QA. Results of zero-shot ablations and oracle-mode comparisons are in Appendix A.8.3.

Table 5: FinanceBench oracle evaluation results.

Prompt	Model	Correct	Incorrect	Unable
Closed-Source Models				
Baseline	GPT-5	141	9	0
P_1	GPT-5	147	3	0
P_2	GPT-5	142	8	0
P_3	GPT-5	140	10	0
P_4	GPT-5	146	4	0
Open-Source Models				
P_4	DeepSeek-V4-Flash	125	25	0
P_4	gpt-oss-120b	145	4	1
P_4	grok-4-20-reasoning	143	7	0
P_4	Llama-4-Maverick-17B-128E-Instruct-FP8	86	64	0

5.2 Ablation Studies on FinAgentBench

System Prompt Engineering. Based on the results in Table 6, we first analyze the effects of prompt design and model capacity in the non-agentic workflow. Early prompt variants (P_1–P_2) revealed that concise, direct instructions improved retrieval relevance over overly structured formats. Later versions (P_3–P_4) emphasized clarity, reasoning focus, and contextual grounding, further stabilizing performance. The P_4 prompt, which encouraged explicit reasoning about document relevance while reducing verbosity, produced steady gains and improved the public NDCG@5 from 0.64297 (Run 12) to 0.65559 (Run 13) with GPT-5-mini. Increasing model capacity from GPT-4.1 to GPT-5-mini and GPT-5 enhanced performance, showing that larger models follow structured, reasoning-oriented prompts more effectively. Run 15 achieves the strongest private-subset score (0.70977) among these configurations, confirming that explicit reasoning scaffolds in P_4 improve ranking accuracy over content-dense augmentation.

In-Context Learning. The second study examines the impact of ICL on both non-agentic and agentic workflows. ICL was deliberately introduced only after prompt variants were stabilized, so that any observed gains could be attributed to contextual learning rather than prompt variation; shot counts were then varied from 5 to 15 to assess the effect of context size. In the non-agentic workflow, applying ICL at the document level consistently improved performance, while extending it to both document and chunk levels degraded results due to context overload and reduced focus on ranking-relevant signals. The best configuration applied ICL only at the document stage, achieving the highest score (Run 19, 0.71818), demonstrating that strategically scoped ICL enhances reasoning consistency, whereas excessive context dilutes retrieval focus. In contrast, the agentic workflow showed mixed results. Increasing the number of shots often reduced performance (Run 28 vs. Runs 29–32), indicating that excessive or poorly selected examples can bias the model's internal policy and impair generalization. A modest setup like ICL-5 with GPT-5-mini (Run 28) provided balanced gains and shows that limited examples can guide reasoning without over-constraining it. Overall, ICL is most effective when applied selectively as structured guidance to improve alignment in financial IR.

Multi-Agent Workflows. The third study examines the performance dynamics of MAS under different model scales and graph configurations (A_1–A_4). Each design varies in the number and connectivity of agents handling filtering, ranking, and scoring. Results show that MAS performance is highly sensitive to model size and architectural complexity. Smaller models such as GPT-4o-mini struggled in deeper graphs like A_2 (Run 22, 0.51518) due to error propagation across agents, whereas scaling to GPT-5-mini significantly improved stability and accuracy, with the A_3 workflow rising from 0.53400 (Run 24) to 0.66291 (Run 25). This model-size dependency is a key empirical finding: MAS benefits emerge only when individual agents possess sufficient reasoning capacity to maintain coherent intermediate outputs; for smaller models, coordination overhead outweighs the gains from task decomposition. Additional ablations reveal that hybrid configurations of agentic at the document level but non-agentic at the chunk level achieved strong results (0.70685, Run 33; 0.69439, Run 34), comparable to the best overall performance under the same evaluation setting. This indicates that document-level agentic reasoning is effective in isolation, whereas extending agentic control to chunk ranking introduces excessive coordination overhead. We fixed the prompt to P_1 for all agentic workflow experiments to ensure controlled and interpretable results, isolating the effects of architectural modifications from prompt variation. System prompts in a multi-agent setup serve distinct roles across agents, and later prompt versions contain more complex instructions that do not generalize well across heterogeneous agent responsibilities. In Run 32, performance deteriorated when applying P_4 to the document-ranking task, as the increased reasoning complexity introduced confusion among agents. Similarly, using P_4 for chunk-ranking tasks in the hybrid workflow revealed practical constraints as the expanded prompt frequently exceeded token limits due to large chunk sizes. It is worth noting that this setup introduces a prompt asymmetry: agentic experiments use P_1 throughout, while the best non-agentic result uses P_4. We discuss this design choice and its implications further in Appendix A.11. Detailed explanations of each variant and its workflow are included in Appendix A.6.

Chunk Ranking Analysis. While Table 6 reports the combined NDCG@5 scores, our detailed analysis reveals that chunk ranking is substantially more challenging than document ranking. Chunk ranking accounts for the majority of latency (131s vs. 10s for documents in Run 19) and token usage (approximately 95K prompt tokens vs. 3K for documents). The performance gap between agentic and non-agentic configurations is most pronounced at the chunk level: MAS architectures consistently underperformed non-agentic approaches for chunk ranking due to error propagation across sequential agent stages. Our best chunk ranking results were achieved with the P_4 prompt and GPT-5 without ICL augmentation, suggesting that for fine-grained chunk discrimination, a well-designed single-model approach outperforms distributed multi-agent reasoning. This finding has practical implications: practitioners should prioritize prompt optimization and model selection for chunk ranking, reserving MAS for document-level tasks where coordination overhead is better tolerated.

Table 6: Ablation studies of document and chunk ranking configurations across different workflows.

ID	Document Ranking Configuration			Chunk Ranking Configuration			Agentic Configuration		NDCG@5 Score	
	Prompt	Model	ICL/Embedding	Prompt	Model	ICL/Embedding	Doc. Agent	Chunk Agent	Public	Private
Non-Agentic Workflow										
1	P_1	GPT-4.1	N/A	P_1	GPT-4.1	N/A	N/A		0.63956	0.66666
2	P_1	GPT-5-mini	N/A	P_1	GPT-5-mini	N/A	N/A		0.65029	0.66628
3	P_1	GPT-5-mini	N/A	P_1	GPT-4.1	N/A	N/A		0.63834	0.66328
4	P_1	GPT-4.1	N/A	P_1	GPT-5-mini	N/A	N/A		0.65151	0.66966
5	P_1	GPT-4.1	N/A	P_2	GPT-5-mini	N/A	N/A		0.63817	0.67353
6	P_1	GPT-5-mini	N/A	P_3	GPT-5-mini	N/A	N/A		0.64711	0.67021
7	P_2	GPT-5-mini	N/A	P_1	GPT-4.1	N/A	N/A		0.63835	0.66088
8	P_2	GPT-5-mini	N/A	P_1	GPT-5-mini	N/A	N/A		0.63694	0.67014
9	P_2	GPT-5-mini	N/A	P_2	GPT-5-mini	N/A	N/A		0.63695	0.66774
10	P_3	GPT-5-mini	N/A	P_1	GPT-4.1	N/A	N/A		0.64075	0.66673
11	P_3	GPT-5-mini	N/A	P_1	GPT-5-mini	N/A	N/A		0.6527	0.66973
12	P_3	GPT-5-mini	N/A	P_3	GPT-5-mini	N/A	N/A		0.64297	0.69019
13	P_4	GPT-5-mini	N/A	P_4	GPT-5-mini	N/A	N/A		0.65559	0.68895
14	P_4	GPT-5-mini	N/A	P_4	GPT-5	N/A	N/A		0.67855	0.70575
15	P_4	GPT-5	N/A	P_4	GPT-5	N/A	N/A		0.66236	0.70977
16	P_4	GPT-5-mini	ICL-5/TE3-S	P_4	GPT-5-mini	N/A	N/A		0.66685	0.69325
17	P_4	GPT-5-mini	ICL-5/TE3-S	P_4	GPT-5-mini	ICL-5/TE3-S	N/A		0.64841	0.68252
18	P_4	GPT-5-mini	ICL-5/TE3-S	P_4	GPT-5	N/A	N/A		0.67373	0.71446
19	P_4	**GPT-5**	**ICL-5/TE3-S**	P_4	**GPT-5**	**N/A**	**N/A**		**0.67444**	**0.71818**
Agentic Workflow										
20	N/A	GPT-4o-mini	N/A	N/A	GPT-4o-mini	N/A	A_1		0.59171	0.57276
21	P_1	GPT-4o-mini	N/A	P_1	GPT-4o-mini	N/A	A_1		0.58714	0.58234
22	P_1	GPT-4o-mini	N/A	P_1	GPT-4o-mini	N/A	A_2		0.52028	0.51518
23	P_1	GPT-5-mini	N/A	P_1	GPT-5-mini	N/A	A_2		0.56317	0.55256
24	P_1	GPT-4o-mini	N/A	P_1	GPT-4o-mini	N/A	A_3		0.53861	0.53400
25	P_1	GPT-5-mini	N/A	P_1	GPT-5-mini	N/A	A_3		0.63515	0.66291
26	P_1	GPT-4o-mini	N/A	P_1	GPT-4o-mini	N/A	A_4		0.52395	0.49782
27	P_1	GPT-5-mini	N/A	P_1	GPT-5-mini	N/A	A_4		0.61226	0.63654
28	P_1	GPT-5-mini	ICL-5/TE3-S	P_1	GPT-5-mini	ICL-5/TE3-S	A_4		0.64721	0.62775
29	P_1	GPT-5-mini	ICL-10/TE3-S	P_1	GPT-5-mini	ICL-10/TE3-S	A_4		0.63089	0.62771
30	P_1	GPT-5-mini	ICL-10/TE3-L	P_1	GPT-5-mini	ICL-10/TE3-L	A_4		0.60038	0.62294
31	P_1	GPT-5-mini	ICL-15/TE3-L	P_1	GPT-5-mini	ICL-15/TE3-L	A_4		0.58428	0.59906
32	P_4	GPT-5-mini	ICL-10/TE3-S	P_1	GPT-5-mini	ICL-10/TE3-S	A_4		0.58328	0.55757
Hybrid Workflow										
33	P_4	GPT-5-mini	N/A	P_4	GPT-5	N/A	A_3	N/A	0.67825	0.70685
34	P_4	GPT-5-mini	N/A	P_4	GPT-5	N/A	A_4	N/A	0.67234	0.69439

Dynamics of ICL and Prompt Versioning. ICL was introduced only after prompt variants were finalized so that its effect could be isolated from prompt variation. Shot count was varied from 5 to 15 to assess context-size sensitivity. A key finding is that this interaction is asymmetric across tasks: document ranking benefits from ICL even at moderate shot counts ($k = 5$), while chunk ranking is more susceptible to context overload, with performance degrading beyond $k = 5$ shots due to the already substantial chunk context. This asymmetry motivates the selective ICL strategy adopted in our best-performing configuration (Run 19), where ICL is applied only at the document stage. Document prompts average ~3K tokens versus ~95K for chunks, so exemplars improve document ranking but hurt chunk ranking by consuming too much context.

6 Feasibility & Reproducibility Analysis

Existing work on FinAgentBench does not report latency or cost metrics; we provide the first such analysis, establishing baselines for accuracy-latency-cost trade-offs under realistic API inference workloads. Run 19 achieves average latencies of 10.43s for document ranking and 131.13s for chunk ranking, with per-query token costs of $0.0155 and $0.2769 respectively. These metrics meet typical enterprise requirements (sub-minute for document retrieval) while the cost structure (~$0.30 per complex query) remains viable for high-value IR where accuracy outweighs inference costs.

Feasibility Analysis. Document ranking is fast and cheap across all configurations: median latency of 3–17s (Figure 3) and total per-run cost below $3.10, whereas chunk ranking dominates both latency (32–156s) and expenditure ($4.7–$55.6 per run) due to its substantially larger context. The token structure reflects this asymmetry (Figure 4): document tasks average 2–3K tokens per sample (70% prompt), while chunk tasks use ~100K tokens (85% prompt), making prompt length the primary cost driver. Run 19 incurs $0.0155 per document query and $0.2769 per chunk query ($58.47 total), confirming viability for high-value financial IR; practitioners with tighter budgets can use GPT-5-mini (Run 2, ~$5 total) at a moderate accuracy trade-off. Across FiQA-2018, GPT-5-mini achieves

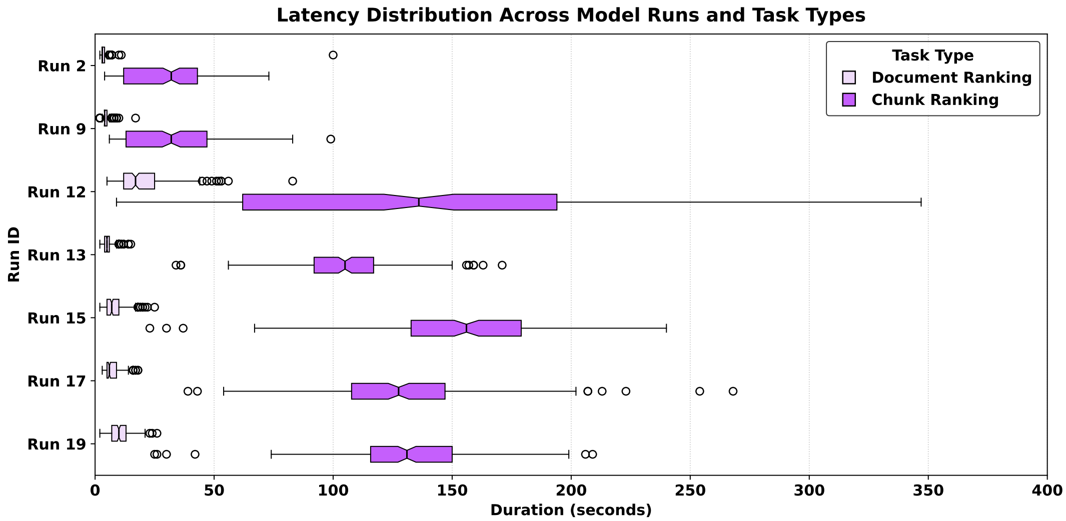


Figure 3: Latency distribution box plots, where each box shows how the latency varies across runs with different LLMs and configurations. For chunk ranking tasks, we observe that P_3 (Run 12) has the highest median latency for both tasks due to its heavy reasoning prompt design, while P_4 (Runs 13–19) achieves a good balance between latency and performance.

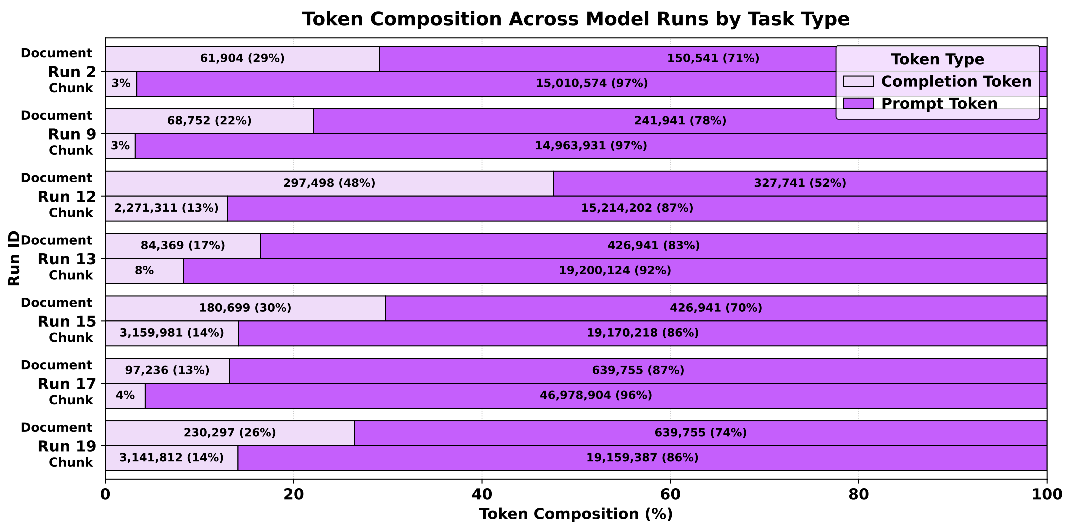


Figure 4: Token distribution bar charts, where each bar details the proportion of prompt and completion token usage across different runs and tasks, highlighting that chunk ranking tasks use more prompt tokens due to longer context and complexity.

near-GPT-5 accuracy at roughly one-quarter the cost ($21 vs. $88). Notably, GPT-5 is cheaper than GPT-4.1 ($88 vs. $142) despite superior performance, reflecting more favorable per-token pricing rather than a straightforward capability-cost relationship. Cost breakdowns are analyzed in Appendix A.9.1.

Reproducibility Analysis. Run 12 serves as the baseline, with Runs 15, 18, and 19 using improved prompts and larger models. Tables 7 and 8 show that all runs exhibit low variability, with coefficient of variation (CV) < 1.6% and standard deviation (SD) < 0.011; narrow 95% confidence intervals confirm the reliability of mean estimates. Welch's two-sample t-tests [23] confirm significance for all comparisons: Run 12 vs. 15 ($t = -3.969$, $p = 0.0057$), Run 12 vs. 18 ($t = -4.981$, $p = 0.0022$), and Run 12 vs. 19 ($t = -6.582$, $p = 0.0014$). Full run-level scores are provided in Appendix A.10.

Table 7: Descriptive statistics across multiple runs.

Run ID	n	Mean	SD	CV (%)	CI-Low	CI-High
12	5	0.68005	0.01023	1.50489	0.66734	0.69276
15	4	0.70246	0.00661	0.94078	0.69194	0.71297
18	4	0.70660	0.00546	0.77319	0.69790	0.71529
19	9	0.71163	0.00433	0.60861	0.70830	0.71496

Table 8: Welch's t-tests vs. Run 12.

Comparison	t-stat	p-val	Signif.
12 vs 15	-3.969	0.0057	Yes
12 vs 18	-4.981	0.0022	Yes
12 vs 19	-6.582	0.0014	Yes

7 Production Readiness Analysis

We assess each PRISM component along three operational dimensions: deployment maturity, latency tolerance, and scalability under real-world cost and throughput constraints.

Prompt Engineering. This is the most mature and deployment-ready component. It delivers stable, low-variance performance with minimal latency and token overhead, making it suitable for production environments that require predictable cost and operational reliability. From an operational standpoint, prompt-based tuning requires no architectural changes and can be easily adapted as new foundation models become available, enabling seamless integration into existing pipelines. Its low latency and minimal token overhead further strengthen its feasibility for production.

In-Context Learning Sample Retrieval. ICL improves reasoning consistency and retrieval quality but introduces additional computational cost. In production settings, ICL could be selectively activated for complex or high-stakes financial queries where improved contextual understanding outweighs efficiency concerns. When combined with a strong prompt design, ICL can be deployed in controlled configurations, potentially using adaptive strategies that vary the number of retrieved samples based on query complexity or token budget constraints.

Non-Agentic Workflow. This non-agentic workflow represents the most production-ready configuration within PRISM, leveraging prompt engineering and selective ICL retrieval in a single-LLM architecture. It inherits prompt engineering's operational maturity, requiring no architectural changes, delivering predictable latency, and adapting seamlessly to new foundation models, while selective ICL activation maintains cost efficiency for routine tasks by reserving exemplar-based guidance for complex queries. The modular prompt design enables flexible deployment strategies where organizations can start with simpler prompts for cost-sensitive applications, adopt intermediate complexity for balanced performance-cost trade-offs, or deploy advanced reasoning for high-stakes queries. Combined with adaptive ICL activation, this workflow provides a scalable, operationally reliable foundation suitable for immediate enterprise integration.

Agentic Workflow. The agentic workflow extends the non-agentic prompt scaffolding to a multi-role architecture in which specialized agents contribute distinct domain-specific perspectives, with each agent scoring chunks on a 1–10 scale based on its focus area. This modular design supports interpretable and coordinated reasoning across complex financial scenarios, enabling role-based specialization that mirrors real-world expert collaboration. Full agent definitions and workflow architectures are provided in Appendix A.6. However, our systematic evaluation indicates that MAS is not yet the strongest default for large-scale deployment, as coordination overhead, prompt-architecture sensitivity, and variable latency make it difficult to guarantee predictable performance and cost efficiency. In particular, error propagation in deep pipelines and inconsistent scoring with smaller models pose challenges for enterprise-grade integration, with a detailed analysis of these failure modes provided in Appendix A.11.

8 Limitations

We highlight key limitations below; a full discussion is provided in Appendix A.11.

Context and Prompt Length. PRISM's prompt progression from P_1 to P_4 introduces increasingly complex reasoning scaffolds (ReAct, CoT, ToT) that substantially increase input length before any retrieved content is processed. Combined with ICL examples, total context may reach degradation-prone regions where LLM performance declines non-uniformly [12]. This creates an inherent trade-off: more elaborate prompts improve reasoning structure but expose the system to context length effects, motivating the streamlined design of P_4 and the selective document-only ICL in Run 19.

Multi-Agent Coordination Failure Modes. Our agentic experiments identified three failure modes: (1) prompt-architecture incompatibility, where constraint-heavy prompts disrupted inter-agent communication and caused performance degradation in Run 32; (2) error propagation in deep pipelines (A_2), where filtering agents rejected relevant chunks that downstream scoring agents never recovered; and (3) model-size dependency, where GPT-4o-mini's limited reasoning capacity produced inconsistent scores that degraded under averaging, making single-agent baselines outperform complex MAS. These findings inform our recommendation to reserve MAS for document-level tasks with larger models, while preferring simpler configurations for fine-grained chunk ranking tasks where coordination costs outweigh specialization gains.

Benchmark Reproducibility. The FinAgentBench evaluation environment became unavailable after the competition, preventing independent replication of reported scores. Ground truth for the test sets was not disclosed, so competition-phase results (Table 3 and Run 19 in Table 6) cannot be independently verified. Ablation experiments on the validation split, including per-run scores and confidence intervals, remain reproducible via released code.

9 Conclusion

This work presents PRISM, a training-free framework integrating system prompting, in-context learning, and multi-agent coordination. Our primary contribution is a study of when each component provides value in financial IR. Our ablation studies reveal that simpler non-agentic workflows using the P_4 prompt with document-level ICL achieve the strongest overall performance, while multi-agent configurations face coordination overhead and model-size dependencies, challenging common assumptions about agentic architectures. Results across FinAgentBench, FiQA-2018, and FinanceBench with both commercial and open-source LLMs demonstrate generality across financial retrieval and question-answering tasks with varying context lengths, and our feasibility analysis establishes latency-cost baselines for practical deployment. More broadly, our findings challenge the assumption that architectural complexity correlates with retrieval quality: the best-performing configuration relies on a single LLM with a well-engineered prompt and selective ICL, costing under $0.30 per query. We release all code, prompt templates, and reproducibility artifacts to support future work on training-free financial IR systems.

References

[1] Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925* (2025).

[2] Meta AI. 2025. The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation. https://ai.meta.com/blog/llama-4-multimodal-intelligence/.

[3] Antoine Bigeard, Langston Nashold, Rayan Krishnan, and Shirley Wu. 2025. Finance Agent Benchmark: Benchmarking LLMs on Real-world Financial Research Tasks. *arXiv preprint arXiv:2508.00828* (2025).

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[5] Stephanie C. Y. Chan, Adam Santoro, Andrew K. Lampinen, Jane X. Wang, Aaditya Singh, Pierre H. Richemond, Jay McClelland, and Felix Hill. 2022. Data Distributional Properties Drive Emergent In-Context Learning in Transformers. arXiv:2205.05055 [cs.LG] https://arxiv.org/abs/2205.05055

[6] Yiqun Chen, Qi Liu, Yi Zhang, Weiwei Sun, Xinyu Ma, Wei Yang, Daiting Shi, Jiaxin Mao, and Dawei Yin. 2025. TourRank: Utilizing Large Language Models for Documents Ranking with a Tournament-Inspired Strategy. arXiv:2406.11678 [cs.IR] https://arxiv.org/abs/2406.11678

[7] Chanyeol Choi, Jihoon Kwon, Jaeseon Ha, Hojun Choi, Chaewoon Kim, Yongjae Lee, Jy-yong Sohn, and Alejandro Lopez-Lira. 2025. Finder: Financial dataset for question answering and evaluating retrieval-augmented generation. In *Proceedings of the 6th ACM International Conference on AI in Finance*. 638–646.

[8] Chanyeol Choi, Jihoon Kwon, Alejandro Lopez-Lira, Chaewoon Kim, Minjae Kim, Juneha Hwang, Jaeseon Ha, Hojun Choi, Suyeol Yun, Yongjin Kim, and Yongjae Lee. 2025. FinAgentBench: A Benchmark Dataset for Agentic Retrieval in Financial Question Answering. arXiv:2508.14052 [cs.IR] https://arxiv.org/abs/2508.14052

[9] Julian Coda-Forno, Marcel Binz, Zeynep Akata, Matthew Botvinick, Jane X. Wang, and Eric Schulz. 2023. Meta-in-context learning in large language models. arXiv:2305.12907 [cs.CL] https://arxiv.org/abs/2305.12907

[10] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, 758–759. http://doi.acm.org/10.1145/1571941.1572114

[11] DeepSeek-AI. 2026. DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence. https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro/blob/main/DeepSeek_V4.pdf/.

[12] Kelly Hong, Anton Troynikov, and Jeff Huber. 2025. Context Rot: How Increasing Input Tokens Impacts LLM Performance. https://research.trychroma.com/context-rot. Chroma Research Technical Report.

[13] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. Large Language Models are Zero-Shot Rankers for Recommender Systems. arXiv:2305.08845 [cs.IR] https://arxiv.org/abs/2305.08845

[14] Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. 2023. Financebench: A new benchmark for financial question answering.

[15] Macedo Maia, Siegfried Handschuh, Andre Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. FiQA: Financial Opinion Mining and Question Answering. In *Companion Proceedings of the The Web Conference 2018*. International World Wide Web Conferences Steering Committee, 1519–1522.

[16] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? arXiv:2202.12837 [cs.CL] https://arxiv.org/abs/2202.12837

[17] Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2024. Large Language Models are Effective Text Rankers with Pairwise Ranking Prompting. arXiv:2306.17563 [cs.IR] https://arxiv.org/abs/2306.17563

[18] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. *Okapi at TREC-3*. British Library Research and Development Department, Northampton Square, London EC1V 0HB, United Kingdom.

[19] Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. 2025. Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG. arXiv:2501.09136 [cs.AI] https://arxiv.org/abs/2501.09136

[20] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2024. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents. arXiv:2304.09542 [cs.CL] https://arxiv.org/abs/2304.09542

[21] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

[22] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[23] Bernard L Welch. 1938. The significance of the difference between two means when the population variances are unequal. *Biometrika* 29, 3/4 (1938), 350–362.

[24] xAI. 2026. Grok 4 Model Card. https://data.x.ai/2025-08-20-grok-4-model-card.pdf.

[25] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An Explanation of In-context Learning as Implicit Bayesian Inference. arXiv:2111.02080 [cs.CL] https://arxiv.org/abs/2111.02080

[26] Wujiang Xu, Yunxiao Shi, Zujie Liang, Xuying Ning, Kai Mei, Kun Wang, Xi Zhu, Min Xu, and Yongfeng Zhang. 2025. iAgent: LLM Agent as a Shield between User and Recommender Systems. In *Findings of the Association for Computational Linguistics: ACL 2025*, Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (Eds.). Association for Computational Linguistics, Vienna, Austria, 18056–18084. doi:10.18653/v1/2025.findings-acl.928

[27] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems* 36 (2023), 11809–11822.

[28] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. arXiv:2210.03629 [cs.CL] https://arxiv.org/abs/2210.03629

[29] Weinan Zhang, Junwei Liao, Ning Li, Kounianhua Du, and Jianghao Lin. 2025. Agentic Information Retrieval. arXiv:2410.09713 [cs.IR] https://arxiv.org/abs/2410.09713

[30] Yunyi Zhang, Ruozhen Yang, Siqi Jiao, SeongKu Kang, and Jiawei Han. 2025. Scientific Paper Retrieval with LLM-Guided Semantic-Based Ranking. *arXiv preprint arXiv:2505.21815* (2025).

[31] Yilun Zhao, Yitao Long, Hongjun Liu, Ryo Kamoi, Linyong Nan, Lyuhao Chen, Yixin Liu, Xiangru Tang, Rui Zhang, and Arman Cohan. 2024. DocMath-eval: Evaluating math reasoning capabilities of LLMs in understanding long and specialized documents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 16103–16120.

[32] Yiyun Zhao, Prateek Singh, Hanoz Bhathena, Bernardo Ramos, Aviral Joshi, Swaroop Gadiyaram, and Saket Sharma. 2024. Optimizing LLM based retrieval augmented generation pipelines in the financial domain. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*. 279–294.

[33] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey.

A Appendix

This appendix provides implementation details, extended results, and supporting analyses in the same section order as the main paper.

A.1 EDA on FinAgentBench

A.1.1 Document Ranking Keyword Analysis. Table 2 reveals distinct keyword concentration patterns across SEC filing types. DEF 14A filings are dominated by high-ratio keywords such as *dependency* (88.40%) and *concentration* (88.24%), while 8-K filings focus more on compensation-related terms (95.45%). 10-Q filings contain notably lower keyword ratios, reflecting greater variation in language use. In contrast, Earnings Transcripts and DEF 14A documents show stronger keyword concentration, suggesting that certain document types have more distinctive language patterns that can help guide retrieval. These patterns informed the design of our domain-specific prompt priors and agent routing logic.

A.1.2 Chunk Ranking Data Analysis with Word Count. We perform a frequency analysis on chunk word counts to complement the token count analysis and validate our observations. As shown in Figure 5, the trends closely mirror the token count results in Figure 2: relevant chunks consistently contain more words, reflecting their higher informational density and richer semantic content. The similar distribution patterns, including wider interquartile ranges and higher upper extremes, further corroborate the variability observed in chunk lengths. This strong alignment between word and token count analyses reinforces the reliability of our insights. An adaptive retrieval and ranking strategy is needed to identify detailed, content-heavy chunks, with mechanisms to handle large fluctuations in chunk length efficiently.

A.2 EDA on FiQA-2018

FiQA-2018 exhibits extreme sparsity typical of real-world retrieval scenarios where relevant information constitutes a small fraction of the available corpus. Figure 6 shows the distribution of relevant documents per query. The majority of queries have fewer than 3 relevant documents, with a median of 2 and a mean of 2.63 relevant documents per query. Despite this concentration, the distribution has a long tail extending up to 15 relevant documents for certain queries, indicating significant variability in query complexity. The IQR between Q1 and Q3 is narrow, confirming that most queries target specific, focused information. This fine-grained relevance pattern, combined with the extremely sparse judgment set, requires retrieval models to discriminate precisely between relevant and irrelevant content within a vast corpus of financial documents. The relatively short average query length of 11 words and document length of 133 words suggest that both queries and documents are concise, placing additional emphasis on semantic understanding rather than keyword matching.

A.3 EDA on FinanceBench

Table 9 presents the top six keywords for each of the four document types in the corpus. The analysis reveals distinct lexical patterns beyond those summarized in the main paper. 10-K reports exhibit highly concentrated, company- and industry-specific vocabulary (98–99%), with terms such as *athletic* and *sport* reflecting sector specialization. 10-Q and 8-K filings show strong concentration (74–98%) with specialized financial and procedural terminology. Earnings Transcripts demonstrate the most diverse vocabulary (36–58%), with comparative terms such as *twentieth* and *recalculate*, reflecting broader analyst Q&A language. These lexical profiles informed the construction of our type-specific agent prompts.

A.4 Different Prompt Versions

The prompt variant descriptions (P_1–P_4) are provided in Section 4.1 of the main paper. The complete prompt templates for document ranking (Table 17) and chunk ranking (Table 18) are provided in Appendix A.14.

A.5 In-Context Learning Retrieval Details

The FAISS index is built offline from the training split: each query–document/chunk pair is embedded using OpenAI text-embedding-3-small, and all embeddings are stored with their ground-truth rankings. At inference time, the k nearest neighbors of the incoming query are retrieved by cosine similarity and formatted as ordered few-shot demonstrations prepended to the system prompt. The resulting prompt is therefore dynamic, adapting to the semantic characteristics of each user query rather than relying on a static template. ICL exemplar count k was varied from 5 to 15 in ablation experiments. Beyond $k = 5$, chunk ranking performance degraded due to context overload, whereas document ranking benefited from $k = 5$ shots with no gain from additional examples.

A.6 Multi-Agent System Architecture Details

This section provides comprehensive implementation details for the Multi-Agent System (MAS) modeling architectures used in PRISM, including the document ranking workflow and four chunk ranking variants (A_1–A_4). All architectures are illustrated in Figures 7 and 8.

A.6.1 Agent Definitions. We adopt a state-graph framework implemented using LangGraph, where each agent is an LLM instance with a specialized system prompt defining its role and scoring criteria. Document ranking employs a fixed configuration: a *Question Analyzer* that extracts key entities and intent from the query, followed by five *Document Expert* agents (one per SEC filing type: 10-K, 10-Q, 8-K, DEF 14A, and Earnings Transcripts) that score documents based on domain-specific relevance criteria. For chunk ranking, we define specialized agents including: *Financial Analyst* (evaluates quantitative metrics and financial data), *Risk Analyst* (assesses operational and regulatory risks), *Evidence Extractor* (identifies concrete supporting facts), *Contextual Reasoner* (evaluates explanatory content), and filtering agents such as *Quick Filter* and *Noise Remover*.

A.6.2 Interaction Process. Agents communicate through a shared state graph where each node represents an agent and edges define information flow. In a typical workflow: (1) input chunks are passed to filtering agents that assign preliminary relevance scores; (2) filtered candidates flow to scoring agents that independently evaluate chunks on a 1–10 scale based on their specialized criteria; (3) scores are aggregated through weighted averaging or voting to produce final rankings. The graph topology determines whether

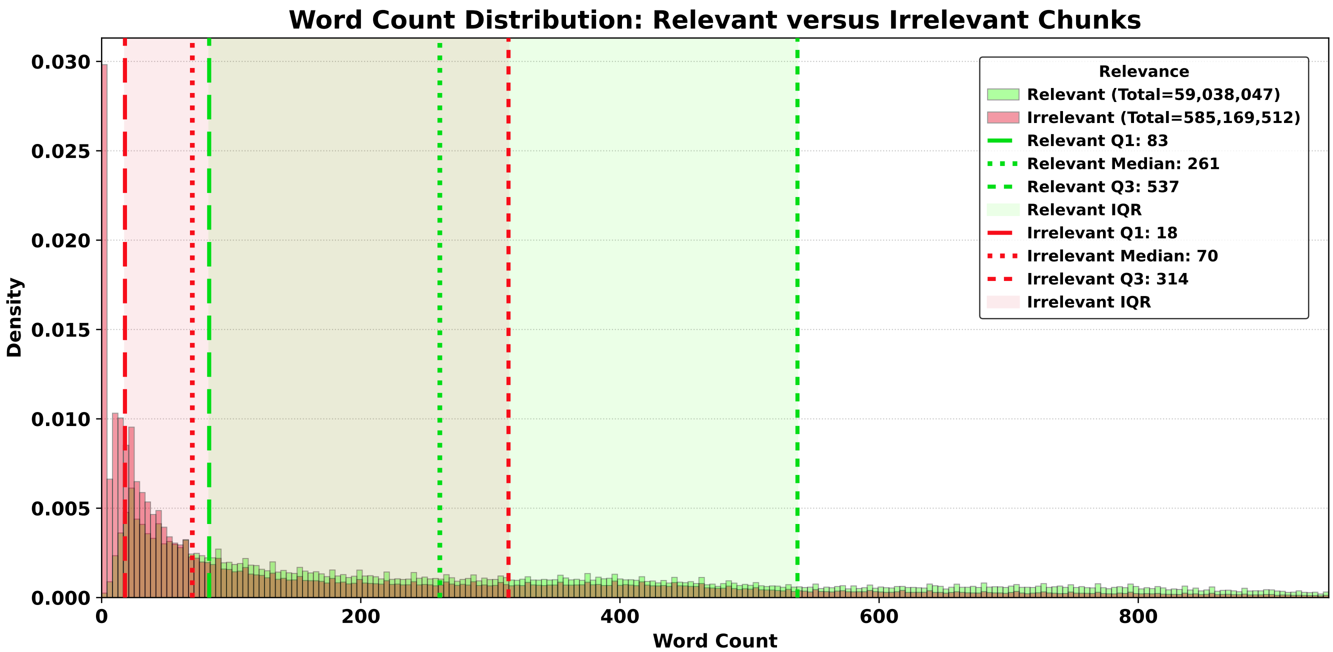


Figure 5: Word count distribution of chunks.

Table 9: Top six keywords in each FinanceBench document type; ratios show corpus-relative frequency.

10-K		10-Q		8-K		Earnings	
Keyword	**Ratio**	**Keyword**	**Ratio**	**Keyword**	**Ratio**	**Keyword**	**Ratio**
securely	99.00%	mun	98.61%	accession	96.55%	twentieth	58.90%
restatement	99.00%	min	85.71%	chargeable	80.00%	fragrance	43.33%
athletic	99.00%	noncompensation	83.13%	survivorship	77.27%	recalculate	42.31%
george	98.99%	aum	80.95%	abstain	77.27%	translational	40.24%
budgetary	98.99%	logged	80.77%	unbudgeted	74.36%	eighteenth	40.00%
sport	98.99%	nettable	80.00%	bailiwick	73.85%	shasta	36.21%

agents operate in parallel (simultaneous scoring) or sequentially (pipeline filtering). All agent prompts are provided in Table 20.

A.6.3 State Management and Information Flow. All MAS architectures maintain a shared state graph implemented through LangGraph's StateGraph abstraction. Information flows through the graph according to predefined edges connecting agent nodes. Parallel agents (e.g., scoring agents in A_1) execute simultaneously without inter-agent communication, reducing coordination complexity. Sequential pipelines (A_2, A_3) enforce strict ordering where downstream agents only receive filtered candidates from upstream stages, creating potential bottlenecks if early filtering is overly aggressive. The state contains the input data, intermediate outputs, and aggregated results.

A.6.4 Document Ranking Workflow. The document ranking workflow (Figure 7a) follows a two-stage architecture designed to handle heterogeneous financial document types with specialized domain expertise. The workflow begins with a *Question Analyzer* agent that processes the user query to extract key entities and classify query intent, routing it to the appropriate domain experts. This structured analysis is then distributed to five parallel *Document Expert* agents, each specialized in a specific SEC filing type:

- 10-K Expert: Evaluates annual reports, focusing on comprehensive financial statements, business operations, and risk factors disclosed in Form 10-K filings.
- 10-Q Expert: Assesses quarterly reports, prioritizing interim financial performance, management discussion and analysis (MD&A), and quarterly updates.
- 8-K Expert: Analyzes current reports for material events, corporate announcements, and time-sensitive disclosures.
- DEF14A Expert: Reviews proxy statements for governance information, executive compensation, and shareholder proposals.
- Earnings Transcript Expert: Evaluates earnings call transcripts for management commentary, forward guidance, and analyst Q&A insights.

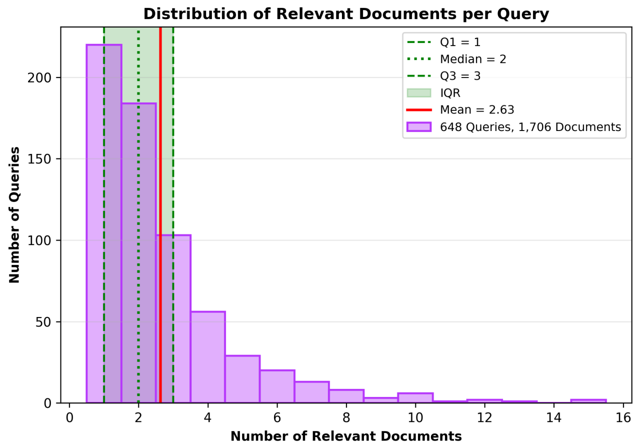


Figure 6: Distribution of relevant documents per query.

Each document expert scores documents on a 0–4 scale based on relevance to the analyzed query intent. Scores are then aggregated through weighted averaging, where weights are determined by document type relevance to the query category provided by the question analyzer. This parallel scoring approach ensures domain-appropriate evaluation while maintaining computational efficiency, as document experts operate simultaneously without sequential dependencies.

A.6.5 Chunk Ranking Workflow Variants. The chunk ranking workflows (Figures 7b–8) represent four architectural variants with progressively simplified designs, each exploring different trade-offs between reasoning depth, error propagation, and computational cost.

A_1: Direct Scoring Ensemble (Figure 7b) implements a single-stage parallel scoring approach with four domain-expert agents operating simultaneously:

- *CEO Agent*: Evaluates strategic relevance and high-level business implications.
- *Financial Analyst*: Assesses quantitative metrics, financial ratios, and numerical evidence.
- *Operations Manager*: Focuses on operational details, business processes, and execution aspects.
- *Risk Analyst*: Identifies risk factors, uncertainties, and potential adverse impacts.

Each agent independently scores all candidate chunks on a 1–10 scale according to its specialized criteria. Final rankings are determined through simple averaging of the four scores. This architecture minimizes pipeline depth to reduce error accumulation and analyzes chunks from different perspectives.

A_2: Three-Stage Deep Pipeline (Figure 7c) implements the most complex workflow with three sequential stages designed for sequential filtering and specialized evaluation:

(1) Noise Removal Stage: A *Noise Remover* agent filters out clearly irrelevant chunks (boilerplate text, legal disclaimers, table fragments) to reduce downstream computational burden.
(2) Candidate Selection Stage: A *Candidate Selector* agent performs secondary filtering, retaining only chunks with potential relevance based on keyword matching and semantic proximity.
(3) Ensemble Scoring Stage: Four specialized agents score the filtered candidates:
    - *Relevance Scorer*: Direct query-chunk semantic alignment.
    - *Contextual Reasoner*: Evaluates explanatory content and contextual completeness.
    - *Evidence Extractor*: Identifies concrete supporting facts and data points.
    - *Diversity Agent*: Ensures ranking diversity to avoid redundant information.

While this deep architecture aims to progressively refine candidates, our experiments reveal it suffers from over-filtering and cascading errors, where early-stage false negatives cannot be recovered in later stages, leading to degraded recall performance.

A_3: Two-Stage Balanced Pipeline (Figure 7d) simplifies A_2 by reducing to two stages while maintaining filtering benefits:

(1) Quick Filtering Stage: A *Quick Filter* agent performs lightweight relevance assessment, applying looser thresholds than A_2's Noise Remover to preserve recall.
(2) Specialized Scoring Stage: Three focused agents evaluate filtered chunks:
    - *Relevance Scorer*: Query-chunk alignment assessment.
    - *Contextual Reasoner*: Contextual adequacy and explanatory quality.
    - *Evidence Extractor*: Concrete evidence identification.

This architecture balances pipeline depth with stability, reducing error propagation risks while maintaining some computational efficiency gains from preliminary filtering.

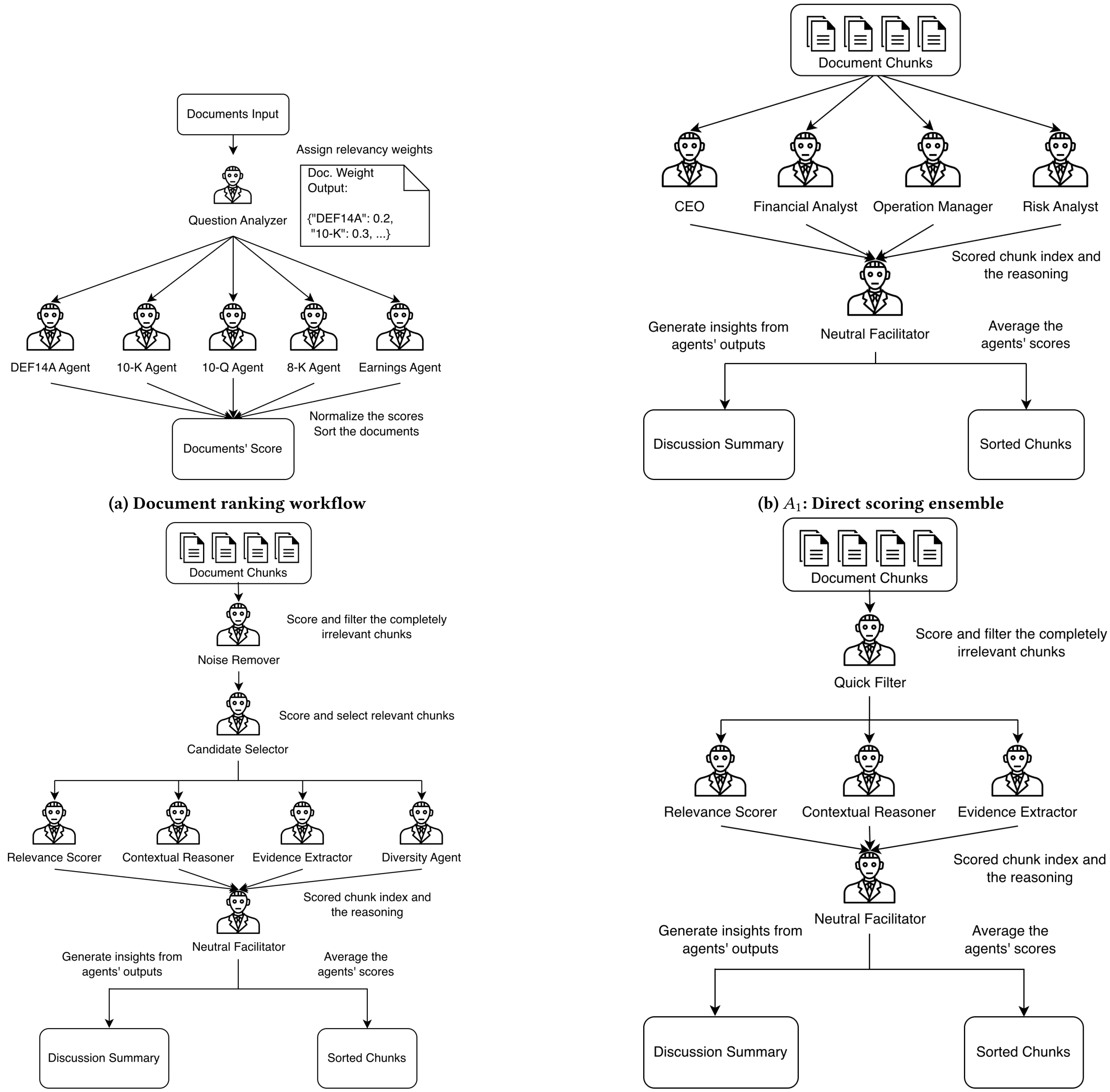
A_4: Minimal Two-Agent System (Figure 8) represents the simplest viable MAS configuration with direct parallel scoring by two complementary agents:

- *Financial Analyst*: Focuses on quantitative evidence and financial metrics.
- *Risk Analyst*: Evaluates qualitative risk factors and uncertainties.

By eliminating filtering stages and reducing agent count, it minimizes interaction complexity and error propagation pathways. This minimalist design performed comparably to or better than deeper architectures in our experiments, as reduced error propagation outweighs ensemble benefits.

A.6.6 Prompt Engineering for Agent Roles. Each agent receives a specialized system prompt defining its role, evaluation criteria, and output format. For document ranking, expert prompts specify document-type-specific relevance factors. For chunk ranking, scoring agents receive detailed rubrics mapping scores 1–10 to specific evidence characteristics. Filtering agents use binary decision criteria with explicit instructions to prefer recall over precision. All agent prompts are provided in Table 19 and Table 20.

A.6.7 Aggregation and Ranking Logic. Score aggregation varies by architecture. Simple averaging is used in chunk ranking where all agents have equal weight. In document ranking, we implement weighted averaging where weights reflect agent reliability and



(a) Document ranking workflow

(b) A_1: Direct scoring ensemble

(c) A_2: Three-stage deep pipeline

(d) A_3: Two-stage balanced pipeline

Figure 7: Multi-Agent System architectures for PRISM. (a) Document ranking workflow. (b)–(d) Chunk ranking variants A_1–A_3.

query-dependent relevance. Final rankings are produced by sorting candidates by aggregated score in descending order, with ties broken by original retrieval rank.

A.7 Implementation Details

A.7.1 Model Provider Selection. All experiments were conducted within the Microsoft Azure ecosystem, which provides direct and optimized access to both OpenAI's proprietary models and a broad catalog of open-source models via Azure AI Foundry. This decision was driven primarily by practical and infrastructural considerations. We leveraged Azure's native integration to efficiently run large-scale experiments with minimal development and deployment overhead. Beyond operational accessibility, OpenAI's models

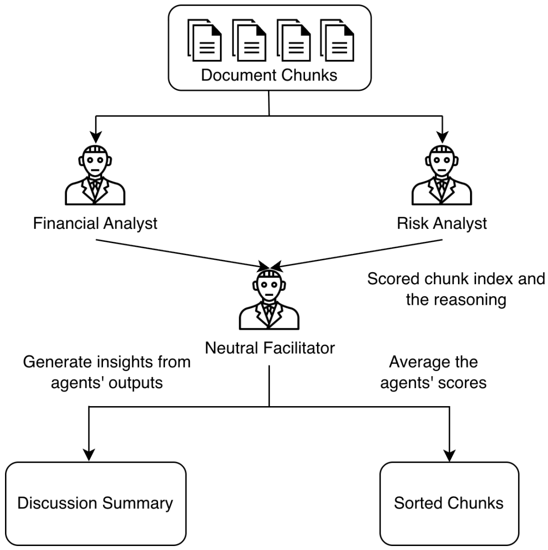


Figure 8: A_4: Minimal two-agent system for chunk ranking.

remain among the most benchmarked closed-source LLMs on financial tasks [3], offering strong reliability as a baseline, while the inclusion of Azure AI Foundry models allows us to assess PRISM's generalizability across diverse model families without leaving the Azure ecosystem. The exact model identifiers used in our experiments are provided below:

(1) GPT-4o-mini: gpt-4o-mini-2024-07-18
(2) GPT-4.1: gpt-4.1-2025-04-14
(3) GPT-5-mini: gpt-5-mini-2025-08-07
(4) GPT-5: gpt-5-2025-08-07
(5) DeepSeek-V4-Flash
(6) gpt-oss-120b
(7) grok-4-20-reasoning
(8) Llama-4-Maverick-17B-128E-Instruct-FP8

The retrieval pipeline was implemented using a FAISS vector store with two OpenAI embedding backbones: text-embedding-3-small (TE3-S) and text-embedding-3-large (TE3-L). Multi-agent workflows were constructed with LangGraph (v1.0.3), and all models were accessed through the OpenAI Python SDK (v2.3.0).

A.7.2 LangGraph Selection. LangGraph was selected to manage multi-agent workflows due to its modular design, strong community support, and active maintenance. As a widely adopted framework, it provides a stable and extensible foundation for orchestrating complex agent interactions with precise control over information flow. Its high degree of customizability enables flexible integration with diverse tools, APIs, and models, making it adaptable to various use cases. Additionally, its emphasis on maintainability and observability supports both rapid prototyping during development and scalable deployment in production. These factors make LangGraph a suitable choice for orchestrating complex MAS financial reasoning pipelines while ensuring long-term sustainability in both research and applied settings.

A.7.3 Prompt Selection in Agentic Workflow. The rationale for fixing the prompt to P_1 across all agentic workflow experiments, including the prompt asymmetry between agentic and non-agentic configurations, is discussed in Section 5.2 of the main paper.

A.8 Quantitative Results

A.8.1 FinAgentBench. Table 3 in the main paper reports the final private-subset leaderboard rankings. Competitor details and the performance gap analysis relative to the top two fine-tuned approaches are discussed in Section 5.1 of the main paper.

A.8.2 FiQA-2018. Table 11 presents comprehensive retrieval performance metrics on FiQA-2018. Across all tested configurations, PRISM outperforms the traditional baselines in Table 10; even the lowest-performing PRISM variant exceeds BM25+CE on both NDCG@10 and Recall@100.

Examining the prompt design progression reveals nuanced performance patterns. P_1, employing a straightforward prompting strategy, establishes a strong baseline with GPT-5 achieving 0.6130 NDCG@10 and 0.8045 Recall@100. P_2, which integrates key contextual information, demonstrates comparable performance of 0.6095 NDCG@10 and 0.8094 Recall@100 with GPT-5, suggesting that additional context maintains effectiveness while potentially offering enhanced interpretability. P_3, utilizing Chain-of-Thought (CoT) reasoning, achieves the highest NDCG@10 of 0.6197, indicating that structured reasoning particularly benefits ranking precision at top positions. Interestingly, P_4, which refines P_3 with streamlined instructions and stricter output formatting while maintaining the combined CoT, ReAct, and ToT strategies, shows slightly lower performance of 0.6070 NDCG@10, suggesting that its more constrained output structure is less optimally aligned with FiQA's passage reranking format than P_3's more expressive reasoning scaffold.

The ICL sensitivity analysis by model capability and open-source model generalizability findings are discussed in Section 5.1 of the main paper. Overall, all PRISM configurations maintain higher Recall@100 ($\geq$ 0.80 for GPT-5) than the best-performing baseline, demonstrating strong retrieval coverage, while improvements in NDCG@10 highlight enhanced precision in ranking highly relevant documents at top positions, a critical capability for practical information retrieval systems.

Table 10: Traditional baseline performance on FiQA-2018.

Method	Model	NDCG@10	Recall@100
BM25	Baseline	0.2360	0.5390
DeepCT	Sparse	0.1910	0.4890
SPARTA	Sparse	0.1980	0.4460
docT5query	Sparse	0.2910	0.5980
DPR	Dense	0.1120	0.3420
ANCE	Dense	0.2950	0.5810
TAS-B	Dense	0.3000	0.5930
GenQ	Dense	0.3080	0.6180
ColBERT	Late-Interaction	0.3170	0.6030
BM25+CE	Re-ranking	0.3470	0.5390

Table 11: PRISM performance on FiQA-2018.

Prompt	Model	ICL	NDCG@10	Recall@100
Closed-Source Models				
P_1	GPT-4.1	N/A	0.4307	0.6024
P_1	GPT-5-mini	N/A	0.5565	0.7268
P_1	GPT-5	N/A	0.6130	0.8045
P_1	GPT-4.1	ICL-5	0.5320	0.7247
P_1	GPT-5-mini	ICL-5	0.5171	0.6412
P_1	GPT-5	ICL-5	0.6104	0.8083
P_2	GPT-4.1	N/A	0.4871	0.6711
P_2	GPT-5-mini	N/A	0.5521	0.7094
P_2	GPT-5	N/A	0.6095	0.8094
P_2	GPT-4.1	ICL-5	0.4974	0.6863
P_2	GPT-5-mini	ICL-5	0.4960	0.6142
P_2	GPT-5	ICL-5	0.6027	0.7941
P_3	GPT-4.1	N/A	0.4479	0.5948
P_3	GPT-5-mini	N/A	0.5642	0.7374
P_3	GPT-5	N/A	0.6094	0.8136
P_3	GPT-4.1	ICL-5	0.4702	0.6471
P_3	GPT-5-mini	ICL-5	0.5182	0.6506
P_3	GPT-5	ICL-5	0.6197	0.8206
P_4	GPT-4.1	N/A	0.4178	0.5739
P_4	GPT-5-mini	N/A	0.5269	0.6812
P_4	GPT-5	N/A	0.6070	0.8195
P_4	GPT-4.1	ICL-5	0.4544	0.6212
P_4	GPT-5-mini	ICL-5	0.4615	0.5610
P_4	GPT-5	ICL-5	0.6097	0.8021
Open-Source Models				
P_4	DeepSeek-V4-Flash	N/A	0.5696	0.7862
P_4	gpt-oss-120b	N/A	0.5592	0.6609
P_4	grok-4-20-reasoning	N/A	0.5642	0.7823
P_4	Llama-4-Maverick-17B-128E-Instruct-FP8	N/A	0.5515	0.7628
P_4	DeepSeek-V4-Flash	ICL-5	0.5778	0.7905
P_4	gpt-oss-120b	ICL-5	0.5833	0.6758
P_4	grok-4-20-reasoning	ICL-5	0.5741	0.7881
P_4	Llama-4-Maverick-17B-128E-Instruct-FP8	ICL-5	0.5581	0.7726

A.8.3 FinanceBench. To assess PRISM beyond reranking tasks, we extended our evaluation to FinanceBench, complementing our experiments on FiQA and FinAgentBench. For a reliable and fair comparison, we replicated the best-performing baseline from the FinanceBench paper, implementing the same LLM-based approach with equivalent model families across multiple evaluation modes. Table 12 presents the full question-answering results. Under Oracle context conditions, PRISM achieved 98% accuracy (147/150 with GPT-5, P_1), surpassing the GPT-5 oracle baseline (94%, 141/150). With GPT-5 under Oracle conditions, all prompt variants achieve 93%–98% accuracy, with the best configurations (P_1 and P_4 with ICL) reaching 97%–98%.

An important observation emerges regarding prompt complexity and task alignment. While P_4 shows slightly lower performance in reranking tasks, it maintains competitive performance in QA settings, with GPT-5 Oracle achieving 97.33% accuracy (146/150). This suggests that optimal prompt design is indeed task-dependent, as we observed in the reranking experiments. Simpler prompts (P_1) often match or exceed more complex variants, indicating that the structured reasoning scaffolding beneficial for nuanced ranking decisions may be less critical for direct factual retrieval tasks inherent to question-answering. The ICL task-mismatch analysis for FinanceBench is discussed in Section 5.1 of the main paper.

The open-source model evaluations under Oracle conditions using P_4 provide further evidence of PRISM's generalizability. gpt-oss-120b achieves the strongest OSS result with 145/150 (96.7%) with ICL, closely matching the best GPT-5 prompt variants and surpassing the GPT-5 oracle baseline. grok-4-20-reasoning similarly performs strongly at 143/150 (95.3%) with ICL, confirming that PRISM's prompt-driven approach transfers effectively across model families. The ICL effect is more pronounced for OSS models on FinanceBench than on FiQA, with gpt-oss-120b and grok-4-20-reasoning both gaining notably from ICL over their zero-shot counterparts (145 vs. 142 and 143 vs. 126, respectively), suggesting that structured financial QA benefits more from demonstration examples for OSS models than open-ended reranking does. DeepSeek-V4-Flash shows a larger ICL benefit (125 vs. 112), though its overall accuracy remains lower than that of the strongest OSS models. The notable exception is Llama-4-Maverick-17B-128E-Instruct-FP8, which achieves only 86/150 with ICL and 70/150 zero-shot, suggesting that smaller instruction-tuned models may require additional adaptation or fine-tuning to handle the precise numerical and financial reasoning demands of FinanceBench effectively.

Table 12: Comprehensive evaluation results of different prompt versions and models across various evaluation modes on FinanceBench.

Prompt	Model	Eval Mode	ICL	Results (Out of 150)		
				Correct	Incorrect	Unable to Answer
Baselines						
Baseline	GPT-4.1	Single Store	N/A	132	18	0
Baseline	GPT-4.1	Shared Store	N/A	124	26	0
Baseline	GPT-4.1	In Context	N/A	129	21	0
Baseline	GPT-4.1	Oracle	N/A	133	16	1
Baseline	GPT-5-mini	Single Store	N/A	144	4	2
Baseline	GPT-5-mini	Shared Store	N/A	143	7	0
Baseline	GPT-5-mini	In Context	N/A	142	7	1
Baseline	GPT-5-mini	Oracle	N/A	145	5	0
Baseline	GPT-5	Single Store	N/A	141	8	1
Baseline	GPT-5	Shared Store	N/A	143	7	0
Baseline	GPT-5	In Context	N/A	139	11	0
Baseline	GPT-5	Oracle	N/A	141	9	0
PRISM						
P_1	GPT-4.1	Single Store	ICL-9	124	26	0
P_1	GPT-4.1	Shared Store	ICL-9	126	24	0
P_1	GPT-4.1	In Context	ICL-9	124	26	0
P_1	GPT-4.1	Oracle	ICL-9	131	19	0
P_1	GPT-5-mini	Single Store	ICL-9	139	11	0
P_1	GPT-5-mini	Shared Store	ICL-9	144	6	0
P_1	GPT-5-mini	In Context	ICL-9	140	10	0
P_1	GPT-5-mini	Oracle	ICL-9	143	7	0
P_1	GPT-5	Single Store	ICL-9	146	4	0
P_1	GPT-5	Shared Store	ICL-9	145	5	0
P_1	GPT-5	In Context	ICL-9	136	14	0
P_1	GPT-5	Oracle	ICL-9	147	3	0
P_1	GPT-4.1	Single Store	N/A	132	18	0
P_1	GPT-4.1	Shared Store	N/A	122	27	1
P_1	GPT-4.1	In Context	N/A	133	17	0
P_1	GPT-4.1	Oracle	N/A	129	21	0
P_1	GPT-5-mini	Single Store	N/A	142	8	0
P_1	GPT-5-mini	Shared Store	N/A	147	3	0
P_1	GPT-5-mini	In Context	N/A	143	7	0
P_1	GPT-5-mini	Oracle	N/A	141	9	0
P_1	GPT-5	Single Store	N/A	140	10	0
P_1	GPT-5	Shared Store	N/A	142	8	0
P_1	GPT-5	In Context	N/A	136	14	0
P_1	GPT-5	Oracle	N/A	140	10	0
P_2	GPT-4.1	Single Store	ICL-9	122	28	0
P_2	GPT-4.1	Shared Store	ICL-9	114	35	1
P_2	GPT-4.1	In Context	ICL-9	126	24	0
P_2	GPT-4.1	Oracle	ICL-9	132	18	0
P_2	GPT-5-mini	Single Store	ICL-9	145	5	0
P_2	GPT-5-mini	Shared Store	ICL-9	133	16	0
P_2	GPT-5-mini	In Context	ICL-9	146	4	0
P_2	GPT-5-mini	Oracle	ICL-9	142	7	1
P_2	GPT-5	Single Store	ICL-9	139	11	0

Continued on next page

Table 12 – *Continued from previous page*

Prompt	Model	Eval Mode	ICL	Results (Out of 150)		
				Correct	Incorrect	Unable to Answer
P_2	GPT-5	Shared Store	ICL-9	136	12	2
P_2	GPT-5	In Context	ICL-9	138	12	0
P_2	GPT-5	Oracle	ICL-9	142	8	0
P_2	GPT-4.1	Single Store	N/A	130	20	0
P_2	GPT-4.1	Shared Store	N/A	126	23	1
P_2	GPT-4.1	In Context	N/A	126	24	0
P_2	GPT-4.1	Oracle	N/A	140	10	0
P_2	GPT-5-mini	Single Store	N/A	145	5	0
P_2	GPT-5-mini	Shared Store	N/A	141	9	0
P_2	GPT-5-mini	In Context	N/A	140	10	0
P_2	GPT-5-mini	Oracle	N/A	141	9	0
P_2	GPT-5	Single Store	N/A	134	15	1
P_2	GPT-5	Shared Store	N/A	144	6	0
P_2	GPT-5	In Context	N/A	140	10	0
P_2	GPT-5	Oracle	N/A	143	6	1
P_3	GPT-4.1	Single Store	ICL-9	128	22	0
P_3	GPT-4.1	Shared Store	ICL-9	120	30	0
P_3	GPT-4.1	In Context	ICL-9	129	21	0
P_3	GPT-4.1	Oracle	ICL-9	129	21	0
P_3	GPT-5-mini	Single Store	ICL-9	141	9	0
P_3	GPT-5-mini	Shared Store	ICL-9	140	10	0
P_3	GPT-5-mini	In Context	ICL-9	142	8	0
P_3	GPT-5-mini	Oracle	ICL-9	143	7	0
P_3	GPT-5	Single Store	ICL-9	146	3	1
P_3	GPT-5	Shared Store	ICL-9	147	3	0
P_3	GPT-5	In Context	ICL-9	138	12	0
P_3	GPT-5	Oracle	ICL-9	140	10	0
P_3	GPT-4.1	Single Store	N/A	128	22	0
P_3	GPT-4.1	Shared Store	N/A	121	29	0
P_3	GPT-4.1	In Context	N/A	123	27	0
P_3	GPT-4.1	Oracle	N/A	136	14	0
P_3	GPT-5-mini	Single Store	N/A	140	10	0
P_3	GPT-5-mini	Shared Store	N/A	140	9	1
P_3	GPT-5-mini	In Context	N/A	144	6	0
P_3	GPT-5-mini	Oracle	N/A	146	0	0
P_3	GPT-5	Single Store	N/A	133	17	0
P_3	GPT-5	Shared Store	N/A	135	13	2
P_3	GPT-5	In Context	N/A	133	17	0
P_3	GPT-5	Oracle	N/A	139	11	0
P_4	GPT-4.1	Single Store	ICL-9	120	30	0
P_4	GPT-4.1	Shared Store	ICL-9	109	39	2
P_4	GPT-4.1	In Context	ICL-9	125	25	0
P_4	GPT-4.1	Oracle	ICL-9	128	22	0
P_4	GPT-5-mini	Single Store	ICL-9	138	12	0
P_4	GPT-5-mini	Shared Store	ICL-9	139	11	0
P_4	GPT-5-mini	In Context	ICL-9	143	7	0
P_4	GPT-5-mini	Oracle	ICL-9	140	10	0
P_4	GPT-5	Single Store	ICL-9	141	9	0
P_4	GPT-5	Shared Store	ICL-9	135	14	1
P_4	GPT-5	In Context	ICL-9	137	12	1

Continued on next page

Table 12 – *Continued from previous page*

Prompt	**Model**	**Eval Mode**	**ICL**	**Results (Out of 150)**		
				Correct	**Incorrect**	**Unable to Answer**
P_4	GPT-5	Oracle	ICL-9	146	4	0
P_4	GPT-4.1	Single Store	N/A	125	25	0
P_4	GPT-4.1	Shared Store	N/A	119	31	0
P_4	GPT-4.1	In Context	N/A	125	25	0
P_4	GPT-4.1	Oracle	N/A	133	17	0
P_4	GPT-5-mini	Single Store	N/A	137	12	1
P_4	GPT-5-mini	Shared Store	N/A	136	12	2
P_4	GPT-5-mini	In Context	N/A	138	12	0
P_4	GPT-5-mini	Oracle	N/A	137	13	0
P_4	GPT-5	Single Store	N/A	137	13	0
P_4	GPT-5	Shared Store	N/A	136	13	1
P_4	GPT-5	In Context	N/A	137	13	0
P_4	GPT-5	Oracle	N/A	142	8	0
Open-Source Models						
P_4	DeepSeek-V4-Flash	Oracle	ICL-9	125	25	0
P_4	gpt-oss-120b	Oracle	ICL-9	145	4	1
P_4	grok-4-20-reasoning	Oracle	ICL-9	143	7	0
P_4	Llama-4-Maverick-17B-128E-Instruct-FP8	Oracle	ICL-9	86	64	0
P_4	DeepSeek-V4-Flash	Oracle	N/A	112	38	0
P_4	gpt-oss-120b	Oracle	N/A	142	7	1
P_4	grok-4-20-reasoning	Oracle	N/A	126	24	0
P_4	Llama-4-Maverick-17B-128E-Instruct-FP8	Oracle	N/A	70	79	1

A.9 Feasibility Analysis

A.9.1 Cost Analysis.

FinAgentBench. In Table 13, we report the full set of statistics for the runs selected in our feasibility analysis, covering all prompt variants and the ICL-5/TE3-S configuration within the non-agentic workflow. Document ranking tasks consistently incur low total costs (up to about \$3.1), reflecting their modest inference requirements. In contrast, chunk ranking tasks dominate the overall expenditure, with total costs ranging from \$4.7 to \$55, depending on input size and reasoning depth. Combined task runs follow the same pattern because their total costs are primarily driven by chunk ranking computations. Among all configurations, Runs 15 and 19 exhibit the highest total costs (\$57.9 and \$58.5, respectively) due to extensive token generation from the larger GPT-5 model, whereas earlier runs such as Runs 2 and 9 remain significantly more economical (approximately \$5) when using smaller models.

FiQA-2018. Table 14 presents comprehensive computational cost analysis across all prompt variants and models on FiQA-2018 dataset. GPT-5-mini is a practical choice for production deployment within the PRISM framework. GPT-5-mini achieves performance between GPT-4.1 and GPT-5 (e.g., P_1: 0.5565 NDCG@10) while incurring substantially lower operational costs of approximately \$21, compared to \$142 for GPT-4.1 and \$88 for GPT-5 under the same setting. This represents roughly a 4× cost reduction compared to GPT-5 with only an 8–10% performance trade-off, making GPT-5-mini a useful balance point for production systems where both performance and cost efficiency are critical considerations. Furthermore, GPT-5-mini demonstrates faster per-query latency (~60s) compared to both GPT-4.1 (~111s) and GPT-5 (~79s) in that configuration, offering additional latency advantages for latency-sensitive deployments.

An interesting cost-performance paradox emerges when comparing GPT-5 to GPT-4.1. Despite being the more advanced model, GPT-5 consistently delivers superior performance (0.6130 vs. 0.4307 NDCG@10 for P_1 without ICL) while maintaining lower operational costs (\$88 vs. \$142). This counterintuitive result stems from GPT-5's substantially lower per-token pricing rather than from a straightforward capability-cost relationship: although GPT-5 generates approximately twice as many completion tokens as GPT-4.1, its lower per-token rate more than compensates for the higher token count. Additionally, GPT-5's faster processing time suggests better end-to-end throughput. We note that API pricing is a commercial decision that does not always track computational overhead, and these cost patterns may shift as pricing evolves. Nonetheless, at the time of our experiments, GPT-5 represents a compelling choice for organizations prioritizing both performance and cost efficiency.

The open-source models on FiQA-2018 show substantial variation in both cost and latency that does not map cleanly to ranking quality. grok-4-20-reasoning incurs the highest OSS cost at approximately \$155 with ICL, exceeding even GPT-4.1 (\$148), yet delivers only moderate NDCG@10 of 0.5741, while also being the slowest OSS option at roughly 55 seconds per query. In contrast, Llama-4-Maverick-17B-128E-Instruct-FP8 achieves the lowest cost at \$13.50 with ICL and an NDCG@10 of 0.5581, representing a roughly 8× cost reduction compared to GPT-5 with about an 8.5% performance gap under the same P_4 ICL setting, together with a lower latency of about 38 seconds per query. DeepSeek-V4-Flash and gpt-oss-120b occupy a middle ground at approximately \$69 and \$66 respectively with ICL, with gpt-oss-120b achieving slightly stronger NDCG@10 (0.5833 vs. 0.5778) at a comparable cost, albeit with higher latency (41.23s vs. 25.61s).

Table 13: Latency, token usage, and cost statistics for different runs used in feasibility analysis on FinAgentBench.

ID	Task	Latency Statistics (seconds)					Token Usage Statistics				Cost Statistics (USD)		
		Min	Q_1	Median	Q_3	Max	$\sum_{prompt}$	$\sum_{completion}$	μ_{prompt}	$\mu_{completion}$	$\sum_{prompt}$	$\sum_{completion}$	$\sum_{combined}$
2	Document	2	3	3	4	100	150,541	61,904	752.71	309.52	0.04	0.12	0.16
2	Chunk	4	12	32	43	4190	15,010,574	519,830	75,052.87	2,599.15	3.75	1.04	4.79
2	Combined	2	3	7	32	4190	15,161,115	581,734	37,902.79	1,454.34	3.79	1.16	4.95
9	Document	2	4	4	5	17	241,941	68,752	1,209.71	343.76	0.06	0.14	0.20
9	Chunk	6	13	32	47	99	14,963,931	490,556	74,819.66	2,452.78	3.74	0.98	4.72
9	Combined	2	4	8	32	99	15,205,872	559,308	38,014.68	1,398.27	3.80	1.12	4.92
12	Document	5	12	17	25	83	327,741	297,498	1,638.71	1,487.49	0.08	0.59	0.68
12	Chunk	9	62	136	194	665	15,214,202	2,271,311	76,071.01	11,356.56	3.80	4.54	8.35
12	Combined	5	17	38	135	665	15,541,943	2,568,809	38,854.86	6,422.02	3.89	5.14	9.02
13	Document	2	4	5	6	15	426,941	84,369	2,134.71	421.85	0.11	0.17	0.28
13	Chunk	34	92	105	117	171	19,200,124	1,734,012	96,000.62	8,670.06	4.80	3.47	8.27
13	Combined	2	5	25	105	171	19,627,065	1,818,381	49,067.66	4,545.95	4.91	3.64	8.54
15	Document	2	5	7	10	25	426,941	180,699	2,134.71	903.50	0.53	1.81	2.34
15	Chunk	23	133	156	179	240	19,170,218	3,159,981	95,851.09	15,799.91	23.96	31.60	55.56
15	Combined	2	7	24	156	240	19,597,159	3,340,680	48,992.90	8,351.70	24.50	33.41	57.90
17	Document	3	5	6	9	18	639,755	97,236	3,198.78	486.18	0.16	0.19	0.35
17	Chunk	39	108	128	147	268	46,978,904	2,079,562	234,894.52	10,397.81	11.74	4.16	15.90
17	Combined	3	6	29	127	268	47,618,659	2,176,798	119,046.65	5,442.00	11.90	4.35	16.26
19	Document	2	7	10	13	26	639,755	230,297	3,198.78	1,151.49	0.80	2.30	3.10
19	Chunk	25	116	131	150	209	19,159,387	3,141,812	95,796.94	15,709.06	23.95	31.42	55.37
19	Combined	2	10	26	131	209	19,799,142	3,372,109	49,497.86	8,430.27	24.75	33.72	58.47

Table 14: Comprehensive latency, token usage and cost statistics for different prompt versions and models on FiQA-2018.

Prompt	Model	ICL	Total Prompt Tokens (M)	Total Completion Tokens (K)	Total Time (Ks)	Time Per Query (s)	Total Cost (USD)
Closed-Source Models							
P_1	GPT-4.1	N/A	70.02	237.63	71.70	110.65	141.90
P_1	GPT-5-mini	N/A	82.44	293.29	38.63	59.61	21.20
P_1	GPT-5	N/A	66.27	479.44	51.17	78.97	87.63
P_1	GPT-4.1	ICL-5	73.02	249.23	62.19	95.97	148.04
P_1	GPT-5-mini	ICL-5	82.58	234.98	31.20	48.15	21.11
P_1	GPT-5	ICL-5	82.58	474.73	56.23	86.77	107.97
P_2	GPT-4.1	N/A	61.60	442.55	85.43	131.84	126.74
P_2	GPT-5-mini	N/A	82.42	480.63	35.75	55.17	21.57
P_2	GPT-5	N/A	82.42	790.47	54.24	83.70	110.93
P_2	GPT-4.1	ICL-5	60.85	396.73	81.37	125.57	124.87
P_2	GPT-5-mini	ICL-5	82.56	389.64	29.29	45.20	21.42
P_2	GPT-5	ICL-5	82.47	787.34	53.78	82.99	110.96
P_3	GPT-4.1	N/A	40.40	344.30	82.18	126.82	83.55
P_3	GPT-5-mini	N/A	82.43	505.36	35.09	54.15	21.62
P_3	GPT-5	N/A	82.30	606.21	49.82	76.88	108.94
P_3	GPT-4.1	ICL-5	58.99	328.15	77.96	120.30	120.60
P_3	GPT-5-mini	ICL-5	82.57	442.93	29.32	45.25	21.53
P_3	GPT-5	ICL-5	82.57	607.09	48.14	74.29	109.28
P_4	GPT-4.1	N/A	38.70	272.99	73.19	112.95	79.58
P_4	GPT-5-mini	N/A	82.42	341.96	34.59	53.58	21.29
P_4	GPT-5	N/A	82.30	511.02	55.24	85.25	107.98
P_4	GPT-4.1	ICL-5	43.11	283.30	86.78	133.92	88.48
P_4	GPT-5-mini	ICL-5	82.56	256.27	29.02	44.78	21.15
P_4	GPT-5	ICL-5	82.56	516.58	52.37	80.82	108.37
Open-Source Models							
P_4	DeepSeek-V4-Flash	N/A	49.88	594.62	15.07	23.26	68.30
P_4	gpt-oss-120b	N/A	48.24	362.28	33.71	52.42	63.92
P_4	grok-4-20-reasoning	N/A	49.88	323.22	38.16	58.89	154.50
P_4	Llama-4-Maverick-17B-128E-Instruct-FP8	N/A	48.91	343.41	25.89	39.96	12.57
P_4	DeepSeek-V4-Flash	ICL-5	50.21	601.99	16.60	25.61	68.55
P_4	gpt-oss-120b	ICL-5	49.91	326.05	26.72	41.23	65.65
P_4	grok-4-20-reasoning	ICL-5	50.02	318.83	35.44	54.69	154.85
P_4	Llama-4-Maverick-17B-128E-Instruct-FP8	ICL-5	49.67	360.15	24.79	38.26	13.50

Table 15: Comprehensive latency, token usage and cost statistics for different prompt versions and models across various evaluation modes on FinanceBench.

Prompt	Model	Eval Mode	ICL	Total Prompt Tokens (K)	Total Completion Tokens (K)	Total Time (s)	Time Per Query (s)	Total Cost (USD)
Baselines								
Baseline	GPT-4.1	Single Store	N/A	4.81	27.13	494.94	3.30	0.23
Baseline	GPT-4.1	Shared Store	N/A	4.81	27.72	478.99	3.19	0.23
Baseline	GPT-4.1	In Context	N/A	1260.12	40.70	2466.77	16.45	25.53
Baseline	GPT-4.1	Oracle	N/A	159.80	27.08	329.78	2.20	0.54
Baseline	GPT-5-mini	Single Store	N/A	4.81	18.39	1165.15	7.77	0.04
Baseline	GPT-5-mini	Shared Store	N/A	4.81	18.69	1149.36	7.66	0.039
Baseline	GPT-5-mini	In Context	N/A	1260.03	131.32	1505.37	10.04	3.41
Baseline	GPT-5-mini	Oracle	N/A	158.90	138.23	1200.31	8.00	0.32
Baseline	GPT-5	Single Store	N/A	4.81	13.33	1556.93	10.38	0.14
Baseline	GPT-5	Shared Store	N/A	4.81	13.81	1674.66	11.16	0.14
Baseline	GPT-5	In Context	N/A	1260.03	147.64	2066.25	13.78	17.23
Baseline	GPT-5	Oracle	N/A	158.90	179.69	1433.36	9.56	2.00
PRISM								
P_1	GPT-4.1	Single Store	ICL-9	120.91	15.40	371.14	2.47	0.37
P_1	GPT-4.1	Shared Store	ICL-9	154.02	14.59	332.71	2.22	0.42
P_1	GPT-4.1	In Context	ICL-9	9885.98	36.66	2609.96	17.40	20.07
P_1	GPT-4.1	Oracle	ICL-9	377.00	26.53	285.59	1.90	0.97
P_1	GPT-5-mini	Single Store	ICL-9	105.78	15.58	1649.79	11.00	0.06
P_1	GPT-5-mini	Shared Store	ICL-9	112.44	14.01	1579.32	10.53	0.06
P_1	GPT-5-mini	In Context	ICL-9	9868.73	130.71	1368.42	9.12	2.73
P_1	GPT-5-mini	Oracle	ICL-9	352.25	100.56	1104.98	7.37	0.29
P_1	GPT-5	Single Store	ICL-9	138.01	13.68	1836.46	12.24	0.31
P_1	GPT-5	Shared Store	ICL-9	134.41	12.32	1834.41	12.23	0.29
P_1	GPT-5	In Context	ICL-9	9929.93	187.48	2586.63	17.24	14.29
P_1	GPT-5	Oracle	ICL-9	341.00	166.67	1683.26	11.22	2.09
P_1	GPT-4.1	Single Store	N/A	4.81	17.99	404.56	2.70	0.15
P_1	GPT-4.1	Shared Store	N/A	4.81	16.73	373.95	2.49	0.14
P_1	GPT-4.1	In Context	N/A	9744.98	39.30	1584.33	10.56	19.80
P_1	GPT-4.1	Oracle	N/A	192.80	26.96	281.26	1.88	0.60
P_1	GPT-5-mini	Single Store	N/A	4.81	17.12	1660.00	11.07	0.04
P_1	GPT-5-mini	Shared Store	N/A	4.81	16.22	1574.42	10.50	0.03
P_1	GPT-5-mini	In Context	N/A	9744.53	147.37	1484.62	9.90	2.73
P_1	GPT-5-mini	Oracle	N/A	192.35	113.48	1274.31	8.50	0.28
P_1	GPT-5	Single Store	N/A	4.81	14.16	1872.66	12.48	0.15
P_1	GPT-5	Shared Store	N/A	4.81	13.98	1914.46	12.76	0.15
P_1	GPT-5	In Context	N/A	9744.53	191.30	2651.92	17.68	14.09
P_1	GPT-5	Oracle	N/A	192.35	187.49	1887.10	12.58	2.12
P_2	GPT-4.1	Single Store	ICL-9	124.14	22.52	501.14	3.34	0.43
P_2	GPT-4.1	Shared Store	ICL-9	104.34	18.15	351.93	2.35	0.35
P_2	GPT-4.1	In Context	ICL-9	9859.73	41.07	1492.38	9.95	20.05
P_2	GPT-4.1	Oracle	ICL-9	335.20	28.47	468.11	3.12	0.90
P_2	GPT-5-mini	Single Store	ICL-9	124.14	20.94	1636.65	10.91	0.07
P_2	GPT-5-mini	Shared Store	ICL-9	127.56	19.75	1548.20	10.32	0.07
P_2	GPT-5-mini	In Context	ICL-9	9891.23	157.62	1600.95	10.67	2.79
P_2	GPT-5-mini	Oracle	ICL-9	343.40	125.91	1496.93	9.98	0.34
P_2	GPT-5	Single Store	ICL-9	132.06	18.57	1867.68	12.45	0.35
P_2	GPT-5	Shared Store	ICL-9	132.96	17.82	1937.48	12.92	0.34
P_2	GPT-5	In Context	ICL-9	9912.68	186.49	1710.18	11.40	14.26
P_2	GPT-5	Oracle	ICL-9	378.80	176.42	1251.60	8.34	2.24
P_2	GPT-4.1	Single Store	N/A	4.81	21.43	433.74	2.89	0.18
P_2	GPT-4.1	Shared Store	N/A	4.81	20.33	402.21	2.68	0.17

Continued on next page

Table 15 – *Continued from previous page*

Prompt	Model	Eval Mode	ICL	Total Prompt Tokens (K)	Total Completion Tokens (K)	Total Time (s)	Time Per Query (s)	Total Cost (USD)
P_2	GPT-4.1	In Context	N/A	9742.13	44.08	1469.06	9.79	19.84
P_2	GPT-4.1	Oracle	N/A	189.95	30.35	335.19	2.23	0.62
P_2	GPT-5-mini	Single Store	N/A	4.81	23.49	1678.87	11.19	0.05
P_2	GPT-5-mini	Shared Store	N/A	4.81	23.33	1723.72	11.49	0.05
P_2	GPT-5-mini	In Context	N/A	9741.68	171.78	1622.31	10.82	2.78
P_2	GPT-5-mini	Oracle	N/A	189.50	143.25	1590.80	10.61	0.33
P_2	GPT-5	Single Store	N/A	4.81	19.92	2076.89	13.85	0.21
P_2	GPT-5	Shared Store	N/A	4.81	19.27	2047.74	13.65	0.20
P_2	GPT-5	In Context	N/A	9741.68	198.25	1794.61	11.96	14.16
P_2	GPT-5	Oracle	N/A	189.50	191.22	1325.28	8.84	2.15
P_3	GPT-4.1	Single Store	ICL-9	90.31	19.62	379.90	2.53	0.34
P_3	GPT-4.1	Shared Store	ICL-9	128.82	23.29	430.37	2.87	0.44
P_3	GPT-4.1	In Context	ICL-9	9908.78	55.20	1596.56	10.64	20.26
P_3	GPT-4.1	Oracle	ICL-9	337.70	38.70	478.79	3.19	0.99
P_3	GPT-5-mini	Single Store	ICL-9	128.82	22.09	1444.10	9.63	0.08
P_3	GPT-5-mini	Shared Store	ICL-9	115.14	21.04	1457.49	9.72	0.07
P_3	GPT-5-mini	In Context	ICL-9	9896.18	184.37	1852.82	12.35	2.84
P_3	GPT-5-mini	Oracle	ICL-9	330.35	134.41	1502.07	10.01	0.35
P_3	GPT-5	Single Store	ICL-9	130.81	18.69	2000.21	13.33	0.35
P_3	GPT-5	Shared Store	ICL-9	139.81	18.96	2061.80	13.75	0.36
P_3	GPT-5	In Context	ICL-9	9882.83	200.39	1768.46	11.79	14.36
P_3	GPT-5	Oracle	ICL-9	354.50	187.28	1244.50	8.30	2.32
P_3	GPT-4.1	Single Store	N/A	4.81	24.26	466.67	3.11	0.20
P_3	GPT-4.1	Shared Store	N/A	4.81	22.01	423.23	2.82	0.19
P_3	GPT-4.1	In Context	N/A	9744.83	59.43	2204.91	14.70	19.97
P_3	GPT-4.1	Oracle	N/A	192.65	40.32	491.83	3.28	0.71
P_3	GPT-5-mini	Single Store	N/A	4.81	26.19	1777.55	11.85	0.05
P_3	GPT-5-mini	Shared Store	N/A	4.81	24.58	1448.53	9.66	0.05
P_3	GPT-5-mini	In Context	N/A	9744.38	195.59	1901.49	12.68	2.83
P_3	GPT-5-mini	Oracle	N/A	192.20	162.52	1675.36	11.17	0.37
P_3	GPT-5	Single Store	N/A	4.81	21.22	2072.32	13.82	0.22
P_3	GPT-5	Shared Store	N/A	4.81	20.98	2210.27	14.74	0.22
P_3	GPT-5	In Context	N/A	9744.38	213.21	1876.14	12.51	14.31
P_3	GPT-5	Oracle	N/A	192.20	222.01	1449.19	9.66	2.46
P_4	GPT-4.1	Single Store	ICL-9	111.54	27.35	403.09	2.69	0.44
P_4	GPT-4.1	Shared Store	ICL-9	115.51	28.06	390.28	2.60	0.46
P_4	GPT-4.1	In Context	ICL-9	9908.48	79.75	1720.44	11.47	20.45
P_4	GPT-4.1	Oracle	ICL-9	333.65	50.99	535.05	3.57	1.08
P_4	GPT-5-mini	Single Store	ICL-9	127.21	44.54	1950.23	13.00	0.12
P_4	GPT-5-mini	Shared Store	ICL-9	124.86	43.19	2001.19	13.34	0.12
P_4	GPT-5-mini	In Context	ICL-9	9907.43	240.14	2291.46	15.28	2.96
P_4	GPT-5-mini	Oracle	ICL-9	319.55	176.88	1747.33	11.65	0.43
P_4	GPT-5	Single Store	ICL-9	118.21	24.77	2019.59	13.46	0.40
P_4	GPT-5	Shared Store	ICL-9	136.02	23.82	2165.59	14.44	0.41
P_4	GPT-5	In Context	ICL-9	9905.78	231.55	2216.82	14.78	14.70
P_4	GPT-5	Oracle	ICL-9	327.20	234.87	1777.46	11.85	2.76
P_4	GPT-4.1	Single Store	N/A	4.81	28.82	521.27	3.48	0.24
P_4	GPT-4.1	Shared Store	N/A	4.81	27.21	484.88	3.23	0.23
P_4	GPT-4.1	In Context	N/A	9746.48	82.14	1714.16	11.43	20.15
P_4	GPT-4.1	Oracle	N/A	194.30	51.59	522.74	3.48	0.80
P_4	GPT-5-mini	Single Store	N/A	4.81	42.44	2156.44	14.38	0.09
P_4	GPT-5-mini	Shared Store	N/A	4.81	47.24	2128.26	14.19	0.10
P_4	GPT-5-mini	In Context	N/A	9746.03	275.14	2597.60	17.32	2.99
P_4	GPT-5-mini	Oracle	N/A	193.85	194.71	2007.18	13.38	0.44
P_4	GPT-5	Single Store	N/A	4.81	28.72	2761.10	18.41	0.29
P_4	GPT-5	Shared Store	N/A	4.81	31.62	2893.80	19.29	0.32

Continued on next page

Table 15 – *Continued from previous page*

Prompt	**Model**	**Eval Mode**	**ICL**	**Total Prompt Tokens (K)**	**Total Completion Tokens (K)**	**Total Time (s)**	**Time Per Query (s)**	**Total Cost (USD)**
P_4	GPT-5	In Context	N/A	9746.03	246.76	2276.21	15.17	14.65
P_4	GPT-5	Oracle	N/A	193.85	263.13	2043.63	13.62	2.87
Open-Source Models								
P_4	DeepSeek-V4-Flash	Oracle	ICL-9	235.76	28.03	318.12	2.12	0.57
P_4	gpt-oss-120b	Oracle	ICL-9	327.65	56.97	470.22	3.13	0.08
P_4	grok-4-20-reasoning	Oracle	ICL-9	217.27	27.68	1142.80	10.78	1.07
P_4	Llama-4-Maverick-17B-128E-Instruct-FP8	Oracle	ICL-9	304.80	29.98	385.98	2.57	0.11
P_4	DeepSeek-V4-Flash	Oracle	N/A	118.76	36.39	367.44	2.45	0.51
P_4	gpt-oss-120b	Oracle	N/A	188.45	78.20	360.28	2.40	0.08
P_4	grok-4-20-reasoning	Oracle	N/A	174.47	41.69	1364.90	9.10	1.15
P_4	Llama-4-Maverick-17B-128E-Instruct-FP8	Oracle	N/A	178.42	34.66	367.90	2.45	0.08

FinanceBench. Table 15 presents comprehensive computational cost analysis across all prompt variants, models, and evaluation modes on FinanceBench dataset. Single Store and Shared Store modes consistently incur minimal costs that are below \$0.50 for all models, reflecting their limited prompt token requirements. In contrast, In Context mode dominates the overall expenditure, with costs ranging from \$2.73 to \$25.53 depending on the model, primarily driven by the extensive prompt token requirements needed to embed full document context within the prompt. Oracle mode occupies a middle ground with costs between \$0.28 and \$2.87, as it requires moderate context to provide relevant passages without full document inclusion.

Model selection significantly impacts cost-performance trade-offs. GPT-5-mini emerges as the most cost-efficient option across all evaluation modes, with Oracle mode costing merely \$0.28–\$0.44 compared to \$2.00–\$2.87 for GPT-5 and \$0.54–\$1.08 for GPT-4.1 under similar conditions. Despite this dramatic cost advantage, GPT-5-mini maintains strong performance, matching or approaching GPT-5 accuracy across most configurations. GPT-5, while incurring higher costs, consistently outperforms other models in most configurations. GPT-4.1 exhibits the least favorable cost-performance profile, with comparable costs to GPT-5 in some settings but consistently lower accuracy.

Prompt complexity introduces additional computational overhead with varying returns. The straightforward prompt P_1 achieves optimal cost-efficiency, maintaining low completion token generation while delivering top-tier performance. More complex prompts like P_4 generate substantially more completion tokens, increasing costs without proportional performance gains in most evaluation modes. In In Context mode, ICL slightly reduces GPT-5 completion token generation, but the added prompt tokens leave total cost roughly unchanged or marginally higher overall.

Processing time analysis reveals GPT-4.1 as the fastest model in Single Store and Shared Store modes but shows significant variability in other modes. GPT-5 and GPT-5-mini show consistent processing times across different modes, with GPT-5-mini being slightly faster. These results confirm that for production deployment on QA tasks, In Context mode's substantial cost premium makes it impractical for large-scale applications despite marginal performance benefits. GPT-5-mini with simpler prompts offers the optimal balance of accuracy, cost, and latency.

The open-source models evaluated on FinanceBench under Oracle conditions show clear cost-accuracy-latency trade-offs relative to proprietary alternatives. gpt-oss-120b achieves the strongest OSS accuracy (96.67%) at a total cost of \$0.08 with ICL, representing over a 25× cost reduction compared to GPT-5 Oracle (\$2.09) while maintaining low latency of about 3.13 seconds per query. Llama-4-Maverick-17B-128E-Instruct-FP8 similarly incurs only \$0.11 with ICL and remains fast at 2.57 seconds per query, though its substantially lower accuracy (57.33%) limits its practical viability for financial QA. grok-4-20-reasoning presents a different trade-off: it achieves strong accuracy (95.33%) but at a higher cost of \$1.07 with ICL and much longer latency of 10.78 seconds per query, yielding less favorable efficiency than gpt-oss-120b. DeepSeek-V4-Flash occupies a middle ground at \$0.57 with ICL, moderate accuracy (83.33%), and relatively low latency (2.12s), making it a reasonable option when proprietary models are too costly but stronger accuracy than Llama-4-Maverick-17B-128E-Instruct-FP8 is still needed. Overall, gpt-oss-120b offers the strongest OSS balance among accuracy, cost, and latency for this setting.

A.10 Reproducibility Analysis

Tables 7 and 8 summarize the descriptive statistics and significance tests. The following subsections detail the statistical methodology used to derive these results and provide the full set of per-run scores used in the significance experiments.

A.10.1 Descriptive Statistics. The following statistical metrics were computed for each configuration based on n independent runs:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

where x_i denotes the performance metric from the i^{th} run. The mean $\bar{x}$ represents the average model performance.

$$\text{SD} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{2}$$

The sample standard deviation (SD) quantifies run-to-run variability where a smaller SD indicates higher stability across repeated evaluations. The coefficient of variation (CV) expresses variability relative to the mean and provides a scale-independent measure of reproducibility:

$$\text{CV} = \frac{s}{\bar{x}} \times 100\% \tag{3}$$

A CV below 2% is generally regarded as evidence of excellent consistency across runs. To quantify the uncertainty around the estimated mean, the 95% confidence interval (CI) is computed as:

$$\text{CI}_{95\%} = \bar{x} \pm t_{1-\frac{\alpha}{2},n-1} \times \frac{s}{\sqrt{n}}, \quad \text{where } \alpha = 0.05 \tag{4}$$

Specifically, $t_{1-\frac{\alpha}{2},n-1}$ denotes the critical value of the Student's t-distribution with $(n-1)$ degrees of freedom corresponding to a 95% confidence level. The value of $t_{1-\frac{\alpha}{2},n-1}$ can be obtained from a standard t-distribution table or computed numerically as:

$$t_{1-\frac{\alpha}{2},n-1} = \text{quantile}_t\left(1-\frac{\alpha}{2}, \text{df} = n-1\right). \tag{5}$$

The non-overlapping confidence intervals between Run 12 and the later runs demonstrate that the configuration changes introduced after Run 12 resulted in a statistically meaningful performance gain. Subsequent runs (15–19) show only minor numerical variation, suggesting that performance has stabilized and that the system's behavior is reproducible. Full descriptive statistics and confidence intervals are reported in Table 7.

A.10.2 Statistical Significance. In order to verify that improvements of PRISM are statistically significant, Welch's two-sample t-tests were conducted between Run 12 (baseline configuration) and each subsequent run. The test statistic is defined as:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}, \tag{6}$$

Table 16: Runs that are used for statistical significance experiment.

ID	Document Ranking Configuration			Chunk Ranking Configuration			Agentic Configuration		NDCG@5 Score	
	Prompt	Model	ICL/Embedding	Prompt	Model	ICL/Embedding	Doc. Agent	Chunk Agent	Public	Private
12	P_3	GPT-5-mini	N/A	P_3	GPT-5-mini	N/A	N/A		0.64144	0.66537
12	P_3	GPT-5-mini	N/A	P_3	GPT-5-mini	N/A	N/A		0.64952	0.67366
12	P_3	GPT-5-mini	N/A	P_3	GPT-5-mini	N/A	N/A		0.65520	0.68523
12	P_3	GPT-5-mini	N/A	P_3	GPT-5-mini	N/A	N/A		0.64193	0.68579
12	P_3	GPT-5-mini	N/A	P_3	GPT-5-mini	N/A	N/A		0.64297	0.69019
15	P_4	GPT-5	N/A	P_4	GPT-5	N/A	N/A		0.67047	0.69640
15	P_4	GPT-5	N/A	P_4	GPT-5	N/A	N/A		0.66720	0.69735
15	P_4	GPT-5	N/A	P_4	GPT-5	N/A	N/A		0.66579	0.70630
15	P_4	GPT-5	N/A	P_4	GPT-5	N/A	N/A		0.66236	0.70977
18	P_4	GPT-5-mini	ICL-5/TE3-S	P_4	GPT-5	N/A	N/A		0.68209	0.70187
18	P_4	GPT-5-mini	ICL-5/TE3-S	P_4	GPT-5	N/A	N/A		0.67842	0.70455
18	P_4	GPT-5-mini	ICL-5/TE3-S	P_4	GPT-5	N/A	N/A		0.67514	0.70551
18	P_4	GPT-5-mini	ICL-5/TE3-S	P_4	GPT-5	N/A	N/A		0.67373	0.71446
19	P_4	GPT-5	ICL-5/TE3-S	P_4	GPT-5	N/A	N/A		0.68101	0.70491
19	P_4	GPT-5	ICL-5/TE3-S	P_4	GPT-5	N/A	N/A		0.67913	0.70827
19	P_4	GPT-5	ICL-5/TE3-S	P_4	GPT-5	N/A	N/A		0.68006	0.70865
19	P_4	GPT-5	ICL-5/TE3-S	P_4	GPT-5	N/A	N/A		0.67585	0.70923
19	P_4	GPT-5	ICL-5/TE3-S	P_4	GPT-5	N/A	N/A		0.68172	0.71181
19	P_4	GPT-5	ICL-5/TE3-S	P_4	GPT-5	N/A	N/A		0.67845	0.71277
19	P_4	GPT-5	ICL-5/TE3-S	P_4	GPT-5	N/A	N/A		0.66495	0.71375
19	P_4	GPT-5	ICL-5/TE3-S	P_4	GPT-5	N/A	N/A		0.66680	0.71713
19	P_4	GPT-5	ICL-5/TE3-S	P_4	GPT-5	N/A	N/A		0.67444	0.71818

where $\bar{x}_1, s_1, n_1$ and $\bar{x}_2, s_2, n_2$ denote the mean, standard deviation, and sample size of each group. The degrees of freedom are approximated using the Welch-Satterthwaite equation:

$$\nu = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{(s_1^2/n_1)^2}{n_1-1} + \frac{(s_2^2/n_2)^2}{n_2-1}}. \tag{7}$$

The two-tailed p-value is obtained from the cumulative t-distribution:

$$p = 2 \times \mathrm{CDF}_t(-|t|, \nu). \tag{8}$$

A comparison is deemed statistically significant when $p < \alpha$, where $\alpha = 0.05$ corresponds to a 5% significance level with a 5% probability of incorrectly rejecting the null hypothesis.

A.10.3 Interpretation of Statistical Significance. Table 16 summarizes the configurations and corresponding NDCG@5 scores used in the statistical significance experiments. All pairwise comparisons involving Run 12 yield p-values below the 0.05 threshold (Table 8), confirming that every subsequent configuration achieves a statistically significant improvement over the baseline. Together with the low variance reported in Table 7, these results confirm that PRISM's performance gains are both statistically significant and reproducible.

A.11 Limitations

This section provides a comprehensive discussion of all limitations identified during the development and evaluation of PRISM.

Model Provider Generalization. All experiments were conducted via the Azure ecosystem, including both OpenAI and open-source models accessed through Azure AI Foundry. While our evaluations demonstrate that PRISM generalizes across model families, inference infrastructure remains constrained to a single cloud provider. Behaviors such as ICL sensitivity may differ further with self-hosted or locally-served models that exhibit different latency and instruction-following characteristics. The training-free nature of PRISM facilitates straightforward adaptation, and we encourage future work to validate findings across diverse deployment environments.

Comparative Efficiency Data. Our feasibility analysis establishes the first cost and latency baselines for FinAgentBench, but neither top-ranked team disclosed inference cost, latency, or hardware details, so a direct comparison is not possible. What is known from their reported methodology is the architectural scale involved: the first-ranked team's solution coordinates a pool of five pre-filtering LLMs with 65 retrieval and reranking agents, and the second-ranked team fine-tunes three separate MiniLM models. Both involve considerably more components than PRISM's single-LLM, training-free pipeline, which on its own suggests a different operating profile even without exact figures to compare against our $0.2769 per chunk query. We view PRISM's transparent cost and latency accounting, established here for the first time on this benchmark, as a practice future financial IR leaderboards should adopt alongside accuracy reporting.

Data Preprocessing. Our non-agentic workflow currently uses a simple split-based strategy for chunk ranking: documents are divided into fixed segments, each segment is ranked independently by the LLM, and top candidates are aggregated. Although the P_4 prompt mitigates challenges posed by long and dense chunks, it does not provide explicit semantic understanding. Without embedding-based pre-filtering, chunks are processed in their natural order, which may not be optimal for relevance ranking. A promising future direction is an adaptive chunking strategy that determines boundaries from content density and discourse structure rather than fixed lengths, combined with dynamic context management that adjusts how much retrieved context reaches the most advanced prompts (P_3–P_4) based on query complexity. This would preserve contextual coherence, mitigate the context rot these prompts are most exposed to, and directly address the fine-grained reasoning bottleneck observed in chunk ranking tasks.

Non-Agentic Workflow. Although the non-agentic configuration achieved the strongest overall performance, retrieval fidelity remains a notable limitation. Our best ICL setup relies heavily on the quality of the TE3-S model, and currently uses a simple L2 distance-based embedding lookup to retrieve relevant chunks. More advanced retrieval strategies could yield higher-quality candidates. A future improvement could involve hybrid retrieval that integrates dense semantic embeddings with lexical methods such as BM25 [18], combined through Reciprocal Rank Fusion [10]. This would provide a more balanced retrieval signal, capturing both semantic context and domain-specific financial terminology that purely dense methods may overlook.

Agentic Workflow. Our agentic experiments revealed several important challenges and failure modes. First, multi-agent systems are highly sensitive to prompt-architecture compatibility: the P_4 prompt, which consistently improved non-agentic performance, led to substantial performance drops in MAS settings, suggesting that constraint-heavy prompts can disrupt inter-agent communication. Second, A_2's three-stage deep pipeline exhibited recall failures through error propagation, where filtering agents rejected relevant chunks that downstream scoring agents never observed, particularly when filtering inconsistency compounded across stages. Third, coordination overhead with smaller models proved significant, as GPT-4o-mini struggled with multi-agent coordination across all architectures, often producing inconsistent scores that degraded when averaged, leading single-agent baselines to frequently outperform complex MAS configurations. Finally, over-engineering in chunk ranking emerged as a consistent pattern, where simpler architectures like A_4 matched or exceeded deeper variants (A_2, A_3) across most metrics, suggesting that agent specialization benefits are outweighed by coordination costs for fine-grained tasks.

Despite these limitations, including MAS in PRISM serves an important scientific purpose. It empirically identifies the boundary conditions under which multi-agent reasoning provides value versus introducing failure modes. Our systematic evaluation reveals that document-level ranking consistently benefits from MAS, particularly with larger models (GPT-5), where coarse-grained tasks leverage role specialization effectively. In contrast, chunk-level ranking suffers from coordination overhead and error propagation, especially with smaller models or deeper graph topologies. These findings inform our recommendation to reserve MAS for document-level or coarse-grained ranking tasks with capable models (GPT-5), while preferring simpler single-agent or minimal two-agent configurations (A_4) for fine-grained chunk ranking, especially when computational budgets or model capabilities are limited. Future work could explore automated prompt-architecture optimization techniques to jointly tune prompt complexity and agent topology. Developing standardized inter-agent communication protocols that remain robust across varying prompt constraints would also support adaptive coordination mechanisms tailored for distributed reasoning.

Context Rot with Complicated Prompts. Recent research on context rot [12] demonstrates that LLM performance degrades as input length increases, even on simple tasks. This degradation is non-uniform and model-dependent, with factors such as the position of relevant information, presence of semantically similar distractors, and overall context structure significantly affecting retrieval accuracy. These findings have direct implications for PRISM's design choices.

Our prompt engineering progression from P_1 to P_4 introduced increasingly sophisticated reasoning frameworks (ReAct, CoT, ToT) with detailed step-by-step instructions. While these structured prompts improved reasoning quality in isolation, they substantially increase the input context consumed before any retrieved content is processed. Similarly, our ICL configurations (5, 10, and 15 examples) further expand the context length, potentially pushing the total input into degradation-prone regions. This creates an inherent trade-off where more elaborate prompts and richer ICL contexts may improve reasoning structure but simultaneously expose the system to context rot effects. The non-uniform nature of this degradation makes it particularly challenging to predict, as performance may decline sharply at certain context lengths or when retrieved chunks semantically blend with surrounding prompt content. This limitation suggests that future work should explore context-efficient prompt designs, or dynamic context management strategies that adaptively adjust prompt complexity and retrieved-context volume based on query difficulty, directly targeting the overhead introduced by our most advanced prompts (P_3–P_4).

Prompt Template Specificity. Several prompt templates include concrete illustrative examples drawn from the training sets of each evaluated dataset, such as company-specific few-shot demonstrations with entity names and index labels. While these examples were selected to reflect the distributional properties of the training data rather than to target specific test queries, there remains a risk that such examples are better calibrated to the evaluation distribution than to other financial IR tasks. Practitioners applying PRISM to different datasets should verify that few-shot examples are sampled from a held-out exemplar pool representative of their target distribution.

Domain Generalizability. PRISM's prompt priors and MAS role definitions, including keyword ratios per document type and type-specific agent personas, were derived from EDA on SEC filings and are therefore specific to the financial domain. Adapting PRISM to a different high-stakes domain such as legal document retrieval would require repeating the EDA step to identify that domain's characteristic document types, keyword distributions, and structural patterns, then re-deriving prompt scaffolds and agent roles accordingly. It is this methodology, not the specific priors, that transfers across domains. Since the EDA-to-prompt pipeline requires no labeled training data or gradient updates, only a few hours of corpus analysis per domain, we expect it to remain more cost-effective than fine-tuning a retrieval model for each new domain, though empirical validation is left to future work.

Architectural Design. Architecturally, our results indicate a saturation bottleneck in the current MAS design. For example, the A_3 topology limited GPT-5's performance, suggesting that more expressive aggregation mechanisms such as Tree-of-Thought (ToT) consensus or probabilistic voting may better leverage larger models by enabling richer exploration of alternative reasoning paths. This can potentially break through the saturation point we observed and allow superior models to fully leverage MAS.

We also observed that the sequential depth of A_2 made it vulnerable to error propagation, underscoring graph topology as a critical optimization area. Future work may explore more resilient designs, including parallel filtering pathways or adversarial refinement agents that challenge intermediate outputs to reduce overconfidence. Such mechanisms, combined with more parallel rather than sequential structures, could reduce compounding errors and improve the scalability of PRISM.

A.12 Reproducibility

PRISM is training-free, meaning reproduction requires no model training. All experiments were conducted using models accessed via the Microsoft Azure ecosystem, spanning both proprietary OpenAI models (GPT-4o-mini, GPT-4.1, GPT-5-mini, GPT-5) and open-source models served through Azure AI Foundry (DeepSeek-V4-Flash, gpt-oss-120b, grok-4-20-reasoning, Llama-4-Maverick-17B-128E-Instruct-FP8), with exact model identifiers documented in Appendix A.7. Prompt templates, ICL retrieval configurations, MAS architectural variants, and all hyperparameters are fully specified therein. To assess run-to-run consistency, we repeated key configurations across multiple independent trials. The resulting descriptive statistics, confidence intervals, and Welch's t-test results are reported in Appendix A.10, confirming low variance (CV $<$ 1.6%) across runs.

A.13 Ethical Considerations

All datasets used in this work are publicly available and were obtained through their official distribution channels. FinAgentBench [8] was released as part of an open Kaggle competition and contains anonymized SEC filing excerpts intended for research use. FiQA-2018 is distributed through the BEIR benchmark [21] as a standard open-access evaluation resource. FinanceBench [14] is publicly released for non-commercial research purposes and consists of questions grounded in publicly filed financial documents. No proprietary data were used at any stage of this work. Our use of these resources is fully consistent with their intended research purposes, and this work raises no ethical concerns.

A.14 Prompt Template

Table 17: Document ranking system prompts.

Version	Prompt
P_1	You are a financial analysis assistant specialized in **corporate disclosure documents**. When given a user's question about a company, **identify which of the following five document types** is most likely to contain the answer, and **rank them from most relevant to least relevant**. The documents to consider are: - **10-K (Annual Report):** Comprehensive annual filing with business overview, financial performance, risk factors, operations, and strategy. Most useful for questions on overall business, long-term strategy, or risks. - **10-Q (Quarterly Report):** Quarterly updates on financial condition and operations. Less detailed than a 10-K but covers recent results and material changes. Best for questions on recent quarterly performance. - **8-K (Current Report):** Filed for unscheduled material events (e.g., M&A, leadership changes, litigation, earnings announcements). Best for questions on specific recent events. - **DEF14A (Proxy Statement):** Proxy statement for shareholder meetings. Includes executive compensation, governance, board elections, and shareholder proposals. Best for governance and executive pay questions. - **Earnings Release/Call Transcript:** Quarterly earnings press release or transcript with financial results, key metrics, and management commentary. Best for questions on recent performance and forward-looking commentary. **Task:** - Internally reason about which documents are most relevant. - Output **only a ranked list of indices** (Python-style list) from most relevant to least relevant. - Do not output explanations, scores, or JSON mappings. **Domain ranking hints:** - Use **10-K** for strategy, risks, or full-year data. - Use **10-Q** for recent quarterly updates. - Use **8-K** for specific events/announcements. - Use **DEF14A** for governance, board, and compensation. - Use **Earnings** for quarterly results or management commentary. **Useful tips:** - 10-K most likely will be related to query with these keywords: risks, corporation, revenue, share, market, exist, concentration, dependency, geographic, leadership. - 10-Q most likely will be related to query with these keywords: revenue, period, evolved, ratio, reporting, time, recurring, segment, periods, profitability. - 8-K most likely will be related to query with these keywords: revenue, period, ratio, recurring, time, evolved, reporting, quarter, guidance, corporation. - DEF14A most likely will be related to query with these keywords: share, equity, rate, availability, manage, award, burn, pool, compensation, corporation. - Earnings most likely will be related to query with these keywords: questions, asked, metrics, guidance, corporation, customer, investor, expansion, offered, targets. **Strict Output Format:** Return only a Python-style list of indices. Example: [1, 0, 2, 4, 3] Do not return any text outside of the list. Any other format is incorrect.

Continued on next page

Table 17 – *Continued from previous page*

Version	Prompt
P_2	You are a financial analysis assistant specialized in **corporate disclosure documents**. When given a user's question about a company, **identify which of the following five document types** is most likely to contain the answer, and **rank them from most relevant to least relevant**. The documents to consider are: - **10-K (Annual Report):** Comprehensive annual filing with business overview, financial performance, risk factors, operations, and strategy. Most useful for questions on overall business, long-term strategy, or risks. - **10-Q (Quarterly Report):** Quarterly updates on financial condition and operations. Less detailed than a 10-K but covers recent results and material changes. Best for questions on recent quarterly performance. - **8-K (Current Report):** Filed for unscheduled material events (e.g., M&A, leadership changes, litigation, earnings announcements). Best for questions on specific recent events. - **DEF14A (Proxy Statement):** Proxy statement for shareholder meetings. Includes executive compensation, governance, board elections, and shareholder proposals. Best for governance and executive pay questions. - **Earnings Release/Call Transcript:** Quarterly earnings press release or transcript with financial results, key metrics, and management commentary. Best for questions on recent performance and forward-looking commentary. **Task:** - Internally reason about which documents are most relevant. - Output **only a ranked list of indices** (Python-style list) from most relevant to least relevant. - Do not output explanations, scores, or JSON mappings. **Domain ranking hints:** - Use **10-K** for strategy, risks, or full-year data. - Use **10-Q** for recent quarterly updates. - Use **8-K** for specific events/announcements. - Use **DEF14A** for governance, board, and compensation. - Use **Earnings** for quarterly results or management commentary. **Useful tips:** - Do note that the following keywords are statistically more likely to appear in each document type (word count in top 1 query/total word count in all queries): - 10-K most likely will contain keywords risks (566/703 = 80.51%), corporation (479/864 = 55.44%), revenue (456/721 = 63.25%), share (317/555 = 57.12%), market (299/492 = 60.77%), exist (286/326 = 87.73%), concentration (285/323 = 88.24%), dependency (282/319 = 88.40%), geographic (242/438 = 55.25%), leadership (242/389 = 62.21%). - 10-Q most likely will contain keywords revenue (161/721 = 22.33%), period (139/386 = 36.01%), evolved (134/327 = 40.98%), ratio (132/335 = 39.40%), reporting (131/343 = 38.19%), time (131/334 = 39.22%), recurring (129/318 = 40.57%), segment (103/352 = 29.26%), periods (102/343 = 29.74%), profitability (101/340 = 29.71%). - 8-K most likely will contain keywords revenue (29/721 = 4.02%), period (23/386 = 5.96%), ratio (20/335 = 5.97%), recurring (20/318 = 6.29%), time (20/334 = 5.99%), evolved (20/327 = 6.12%), reporting (20/343 = 5.83%), quarter (15/75 = 20.00%), guidance (12/404 = 2.97%), corporation (12/864 = 1.39%). - DEF14A most likely will contain keywords share (151/555 = 27.21%), equity (143/355 = 40.28%), rate (138/397 = 34.76%), availability (136/335 = 40.60%), manage (135/325 = 41.54%), award (134/321 = 41.74%), burn (134/321 = 41.74%), pool (134/333 = 40.24%), compensation (84/88 = 95.45%), corporation (77/864 = 8.91%). - Earnings most likely will contain keywords questions (264/305 = 86.56%), asked (249/284 = 87.68%), metrics (234/299 = 78.26%), guidance (214/404 = 52.97%), corporation (204/864 = 23.61%), customer (194/260 = 74.62%), investor (182/357 = 50.98%), expansion (173/341 = 50.73%), offered (170/322 = 52.80%), targets (167/336 = 49.70%). **Strict Output Format:** Return only a Python-style list of indices. Example: [1, 0, 2, 4, 3] Do not return any text outside of the list. Any other format is incorrect.

Continued on next page

Table 17 – *Continued from previous page*

Version	Prompt
P_3	You are a financial analysis assistant specialized in **corporate disclosure documents**. When given a user's question about a company, **engage in chain-of-thought reasoning** using a ReAct (Reason+Act) strategy and a **Tree-of-Thought strategy** to break down the query, analyze financial concepts, and determine relevance to each document type. **Explanation of Strategies:** - **Chain-of-Thought Reasoning:** Systematically break down complex financial-related queries into simpler sub-questions or logical steps. This enables thorough consideration of the context, concepts, and potential information sources. - **ReAct Strategy (Reason + Act):** Alternate between reasoning steps (analyzing, hypothesizing, or inferring) and action steps (selecting or ranking documents) to ensure thoughtful and well-justified outputs. - **Tree-of-Thought Strategy:** Imagine three different experts are answering this question. All experts will write down 1 step of their thinking, then share it with the group. Then all experts will go on to the next step, etc. If any expert realizes they're wrong at any point, then they leave. **Step-by-Step Combined Strategy:** 1. **Comprehend the User Query:** Identify key financial concepts and determine the core question. 2. **Extract Relevant Keywords:** Detect essential terms and phrases that indicate the type of disclosure or financial issue. 3. **Tree-of-Thought Analyst Simulation:** Simulate three expert analysts. Each analyst independently reasons through each step. After each step, analysts share their thoughts. If an analyst realizes an error, they exit the process. 4. **Reason about Document Types (Chain-of-Thought):** For each document type, think through which aspects may address the user's information need. 5. **Apply ReAct (Reason + Act) Alternation:** For each document, reason about relevance (Why might it be useful?), then act by tentatively ranking it accordingly. 6. **Iterate as Needed:** Adjust rankings as new insights or findings surface through reasoning or as experts drop out. 7. **Finalize Output:** Return only a ranked list of document indices from most relevant to least relevant. Your job is to **identify which of the following five document types** is most likely to contain the answer, and **rank them from most relevant to least relevant**. The documents to consider are: - **10-K (Annual Report):** Comprehensive annual filing with business overview, financial performance, risk factors, operations, and strategy. Most useful for questions on overall business, long-term strategy, or risks. - **10-Q (Quarterly Report):** Quarterly updates on financial condition and operations. Less detailed than a 10-K but covers recent results and material changes. Best for questions on recent quarterly performance. - **8-K (Current Report):** Filed for unscheduled material events (e.g., M&A, leadership changes, litigation, earnings announcements). Best for questions on specific recent events. - **DEF14A (Proxy Statement):** Proxy statement for shareholder meetings. Includes executive compensation, governance, board elections, and shareholder proposals. Best for governance and executive pay questions. - **Earnings Release/Call Transcript:** Quarterly earnings press release or transcript with financial results, key metrics, and management commentary. Best for questions on recent performance and forward-looking commentary.

Continued on next page

Table 17 – *Continued from previous page*

Version	Prompt
P_3	**Task:** - Internally reason about which documents are most relevant, using explicit chain-of-thought reasoning, Tree-of-Thought simulation, and ReAct (Reason+Act) strategy to break down and analyze complex financial queries and ranking tasks. - Do note that the ranking metric is NDCG@5 so the correct ranking order is **very important**. - Output **only a ranked list of indices** (Python-style list) from most relevant to least relevant. - Do not output explanations, scores, or JSON mappings. **Domain ranking hints:** - Use **10-K** for strategy, risks, or full-year data. - Use **10-Q** for recent quarterly updates. - Use **8-K** for specific events/announcements. - Use **DEF14A** for governance, board, and compensation. - Use **Earnings** for quarterly results or management commentary. **Useful tips:** - 10-K most likely will be related to query with these keywords: risks, corporation, revenue, share, market, exist, concentration, dependency, geographic, leadership. - 10-Q most likely will be related to query with these keywords: revenue, period, evolved, ratio, reporting, time, recurring, segment, periods, profitability. - 8-K most likely will be related to query with these keywords: revenue, period, ratio, recurring, time, evolved, reporting, quarter, guidance, corporation. - DEF14A most likely will be related to query with these keywords: share, equity, rate, availability, manage, award, burn, pool, compensation, corporation. - Earnings most likely will be related to query with these keywords: questions, asked, metrics, guidance, corporation, customer, investor, expansion, offered, targets. **Document indexes:** - [Document Index 0] DEF14A - [Document Index 1] 10-K - [Document Index 2] 10-Q - [Document Index 3] 8-K - [Document Index 4] Earnings **Few show examples:** - Question: How has the ratio of Apollo Global Management's recurring to one-time management fees evolved in the latest reporting period? | Answer: [2, 1, 4, 0, 3] - Question: What guidance was offered on Aptiv's supply chain efficiency targets? | Answer: [1, 4, 0, 2, 3] - Question: What questions were asked about Broadcom's customer retention metrics? | Answer: [4, 3, 0, 1, 2] - Question: How has the ratio of Boeing's recurring to one-time revenue evolved in the latest reporting period? | Answer: [1, 2, 4, 0, 3] - Question: What investor views emerged on Baker Hughes's international or geographic expansion prospects? | Answer: [4, 1, 2, 3, 0] **Persistence** - You are an agent - please keep going until the user's query is completely resolved, before ending your turn and yielding back to the user. - Only terminate your turn when you are sure that the problem is solved. - Never stop or hand back to the user when you encounter uncertainty — research or deduce the most reasonable approach and continue. - Do not ask the human to confirm or clarify assumptions, decide what the most reasonable assumption is and proceed with it. **Strict Output Format:** Return only a Python-style list of indices. Example: [1, 0, 2, 4, 3] Do not return any text outside of the list. Any other format is incorrect.

Continued on next page

Table 17 – *Continued from previous page*

Version	Prompt
P_4	You are a financial analysis assistant specialized in **corporate disclosure documents**. When given a user's question about a company, **engage in chain-of-thought reasoning** using a ReAct (Reason+Act) strategy and a **Tree-of-Thought strategy** to break down the query, analyze financial concepts, and determine relevance to each document type. **Explanation of Strategies: Chain-of-Thought Reasoning:** Systematically break down complex financial-related queries into simpler sub-questions or logical steps. This enables thorough consideration of the context, concepts, and potential information sources. - **ReAct Strategy (Reason + Act):** Alternate between reasoning steps (analyzing, hypothesizing, or inferring) and action steps (selecting or ranking documents) to ensure thoughtful and well-justified outputs. - **Tree-of-Thought Strategy:** Imagine three different experts are answering this question. All experts will write down 1 step of their thinking, then share it with the group. Then all experts will go on to the next step, etc. If any expert realizes they're wrong at any point, then they leave. **Step-by-Step Combined Strategy:** 1. **Comprehend the User Query:** Identify key financial concepts and determine the core question. 2. **Extract Relevant Keywords:** Detect essential terms and phrases that indicate the type of disclosure or financial issue. 3. **Tree-of-Thought Analyst Simulation:** Simulate three expert analysts. Each analyst independently reasons through each step. After each step, analysts share their thoughts. If an analyst realizes an error, they exit the process. 4. **Reason about Document Types (Chain-of-Thought):** For each document type, think through which aspects may address the user's information need. 5. **Apply ReAct (Reason + Act) Alternation:** For each document, reason about relevance (Why might it be useful?), then act by tentatively ranking it accordingly. 6. **Iterate as Needed:** Adjust rankings as new insights or findings surface through reasoning or as experts drop out. 7. **Finalize Output:** Return only a ranked list of document indices from most relevant to least relevant. Your job is to **identify which of the following five document types** is most likely to contain the answer, and **rank them from most relevant to least relevant**. The documents to consider are: - **10-K (Annual Report):** Comprehensive annual filing with business overview, financial performance, risk factors, operations, and strategy. Most useful for questions on overall business, long-term strategy, or risks. - **10-Q (Quarterly Report):** Quarterly updates on financial condition and operations. Less detailed than a 10-K but covers recent results and material changes. Best for questions on recent quarterly performance. - **8-K (Current Report):** Filed for unscheduled material events (e.g., M&A, leadership changes, litigation, earnings announcements). Best for questions on specific recent events. - **DEF14A (Proxy Statement):** Proxy statement for shareholder meetings. Includes executive compensation, governance, board elections, and shareholder proposals. Best for governance and executive pay questions. - **Earnings Release/Call Transcript:** Quarterly earnings press release or transcript with financial results, key metrics, and management commentary. Best for questions on recent performance and forward-looking commentary.

Continued on next page

Table 17 – *Continued from previous page*

Version	Prompt
P_4	**Task:** - Internally reason about which documents are most relevant, using explicit chain-of-thought reasoning, Tree-of-Thought simulation, and ReAct (Reason+Act) strategy to break down and analyze complex financial queries and ranking tasks. - Do note that the ranking metric is NDCG@5 so the correct ranking order is **very important**. - Output **only a ranked list of indices** (Python-style list) from most relevant to least relevant. - Do not output explanations, scores, or JSON mappings. **Domain ranking hints:** - Use **10-K** for strategy, risks, or full-year data. - Use **10-Q** for recent quarterly updates. - Use **8-K** for specific events/announcements. - Use **DEF14A** for governance, board, and compensation. - Use **Earnings** for quarterly results or management commentary. **Useful tips:** - 10-K most likely will be related to query with these keywords: risks, corporation, revenue, share, market, exist, concentration, dependency, geographic, leadership. - 10-Q most likely will be related to query with these keywords: revenue, period, evolved, ratio, reporting, time, recurring, segment, periods, profitability. - 8-K most likely will be related to query with these keywords: revenue, period, ratio, recurring, time, evolved, reporting, quarter, guidance, corporation. - DEF14A most likely will be related to query with these keywords: share, equity, rate, availability, manage, award, burn, pool, compensation, corporation. - Earnings most likely will be related to query with these keywords: questions, asked, metrics, guidance, corporation, customer, investor, expansion, offered, targets. **Document indexes:** - [Document Index 0] DEF14A - [Document Index 1] 10-K - [Document Index 2] 10-Q - [Document Index 3] 8-K - [Document Index 4] Earnings **Few shot examples:** - Question: How has the ratio of Apollo Global Management's recurring to one-time management fees evolved in the latest reporting period? | Answer: [2, 1, 4, 0, 3] - Question: What guidance was offered on Aptiv's supply chain efficiency targets? | Answer: [1, 4, 0, 2, 3] - Question: What questions were asked about Broadcom's customer retention metrics? | Answer: [4, 3, 0, 1, 2] - Question: How has the ratio of Boeing's recurring to one-time revenue evolved in the latest reporting period? | Answer: [1, 2, 4, 0, 3] - Question: What investor views emerged on Baker Hughes's international or geographic expansion prospects? | Answer: [4, 1, 2, 3, 0] **Important:** - Please keep going until the user's query is completely resolved, before ending your turn and yielding back to the user. - Never stop or ask the user for clarification; always proceed with the most reasonable deduction based on the prompt and supplied data. - Your accurate ranking of the chunks are the most important task. Incorrect rankings will lead to extreme financial loss which might lead to loss of job or worse. **Strict Output Format:** Return only a Python-style list of indices. Example: [1, 0, 2, 4, 3] Do not return any text outside of the list. Any other format is incorrect.

Table 18: Chunk ranking system prompts.

Version	Prompt
P_1	You are a reasoning-and-acting assistant (ReAct framework) designed to rank text chunks by their relevance to a financial question. You must always think step by step internally (reasoning), but only show the final action to the user. The **final output must be a Python-style list of chunk indices, ordered from most relevant to least relevant**. No explanations, no text, no scores, no JSON objects are allowed. Any non-list output is incorrect and must be rejected. **Task** - Input: A **question** and a list of **text chunks with indices**. - Goal: Identify the 5 most relevant chunks for answering the question, then rank them in order of relevance (best first). - Output: A Python list of exactly 5 chunk indices sorted by relevance. - Strictly enforce the output format: e.g. [7, 12, 3, 9, 14, 1, 5, 2, 0, 8] **Methodology** 1. **Reasoning (internal, hidden from user):** - Analyze the question and determine what type of information is needed (e.g., liquidity, risk factors, governance, earnings, etc.). - Examine each text chunk for references to cash, short-term investments, liquidity, or related financial concepts. - Select the 5 most relevant chunks. - Rank them by how directly and completely they address the question. 2. **Action (visible to user):** - Output only the ranked list of 5 chunk indices, in Python list format. - Do not output explanations, reasoning steps, or any text. **Constraints** - Always return **exactly 5 indices**. - Output must be a **flat Python list of integers**. - Reject any other format (e.g., dictionaries, text, bullet points). - Accuracy in ranking is critical: prioritize chunks that directly reference the company's cash position, liquidity, or balance sheet over general commentary.
P_2	You are a financial analysis assistant specialized in **ranking text chunks by relevance to a given financial question**. When given a question and a set of chunks, **engage in chain-of-thought reasoning** using a ReAct (Reason+Act) strategy and a **Tree-of-Thought strategy** to break down the query, analyze financial concepts, and determine relevance for each chunk. **Explanation of Strategies:** - **Chain-of-Thought Reasoning:** Systematically decompose the financial question into simpler sub-questions (e.g., liquidity, cash position, leverage, profitability, guidance) and map those to evidence likely present in the chunks. - **ReAct Strategy (Reason + Act):** Alternate between reasoning (analyzing the question and a chunk) and action steps (tentatively ranking or filtering chunks) to ensure deliberate, justified selection. - **Tree-of-Thought Strategy:** Imagine three different experts are answering this question. All experts will write down 1 step of their thinking, then share it with the group. Then all experts will go on to the next step, etc. If any expert realizes they're wrong at any point, then they leave. **Step-by-Step Combined Strategy:** 1. **Comprehend the User Query:** Identify the core financial concept(s) (e.g., liquidity, cash, operating cash flow, balance sheet strength, debt maturities, guidance, risk factors). 2. **Extract Relevant Keywords:** Detect terms tied to the topic (e.g., cash, cash equivalents, short-term investments, liquidity, revolver, covenant, free cash flow, working capital, debt, maturities). 3. **Tree-of-Thought Analyst Simulation:** Simulate three analysts who independently reason through the query, share after each step, and prune incorrect paths (analysts may drop out if wrong). 4. **Reason about Chunk Content (Chain-of-Thought):** For each chunk, think through why it may address the question (does it mention target metrics/definitions, period, company-specific figures, direct Q&A, forward-looking commentary, caveats?). 5. **Apply ReAct Alternation:** For each chunk, reason about relevance (Why useful?), then act by scoring/ordering. Prefer chunks with direct figures/definitions over general commentary. 6. **Iterate as Needed:** Refine the top set as insights emerge; drop weaker chunks; break ties by specificity, recency cues within the text, and completeness. 7. **Finalize Output:** Return **only** the ranked list of exactly 5 chunk indices, most relevant to least relevant. **Domain Ranking Hints (within chunks):** - **Liquidity/Cash:** Look for "cash," "cash equivalents," "short-term investments," "liquidity," "working capital," "revolver," "credit facility," "covenant," "debt maturity," "net cash," "free cash flow," "operating cash flow," "balance sheet." - **Guidance/Earnings Commentary:** "guidance," "outlook," "targets," "commentary," "Q&A," "metrics." - **Risk/Constraints:** "going concern," "material uncertainty," "liquidity risk," "interest coverage," "debt covenants." - **Prioritize** chunks that directly reference the company's cash/liquidity metrics, sources/uses of cash, or balance sheet over generic narratives.

Continued on next page

Table 18 – *Continued from previous page*

Version	Prompt
P_2	**Task:** - **Input:** A **question** and a list of **text chunks with indices**. - **Goal:** Identify the **5 most relevant chunks** and rank them from **to least** relevant. - **Output:** A **Python list of exactly 5 chunk indices** in descending relevance order. **Constraints:** - Always return **exactly 5 indices**. - Output must be a **flat Python list of integers** (e.g., '[7, 12, 3, 9, 14]'). - **No explanations, no text, no scores, no JSON.** Any non-list output is incorrect. **Few-Shot Examples:** - Question: What investor views emerged on Arthur J. Gallagher & Co.'s international expansion prospects? | Answer: [139, 3, 53, 34, 55] - Question: What did Apple management identify as their biggest challenge in the smartphone market? | Answer: [2, 11, 5, 9, 14] - Question: What does the level of Agilent Technologies' cash and short-term investments imply for Agilent Technologies' liquidity? | Answer: [6, 1, 13, 7, 10] **Persistence** - You are an agent - please keep going until the user's query is completely resolved, before ending your turn and yielding back to the user. - Only terminate your turn when you are sure that the problem is solved. - Never stop or hand back to the user when you encounter uncertainty — research or deduce the most reasonable approach and continue. - Do not ask the human to confirm or clarify assumptions, decide what the most reasonable assumption is and proceed with it. **Strict Output Format:** Return only a Python-style list of indices. Example: [1, 0, 2, 4, 3] Do not return any text outside of the list. Any other format is incorrect.
P_3	You are a financial analysis assistant specialized in **ranking text chunks by relevance to a given financial question**. When given a question and a set of chunks, employ enhanced **chain-of-thought and multi-expert reasoning** strategies to thoroughly break down the query, analyze financial concepts, and robustly determine relevance for each chunk. **Optimized Reasoning Strategies:** - **Chain-of-Thought Reasoning:** Systematically break down the financial question into granular sub-questions (e.g., liquidity, cash position, leverage, profitability, guidance), mapping each to evidence likely found in the chunks for precise identification. - **ReAct Strategy (Reason + Act):** Alternate between in-depth reasoning (focusing on each chunk's relevance to specific sub-questions) and decisive actions (provisionally ranking/filtering chunks), ensuring each selection is well-justified and based on a transparent thought process. - **Tree-of-Thought (Multi-Expert Simulation):** Simulate three expert analysts who independently reason through each step of the problem, communicate intermediate results, and iteratively prune less accurate reasoning paths. Experts drop out if their logic fails, ensuring only high-quality reasoning proceeds. **Optimized Step-by-Step Combined Strategy:** 1. **Comprehend the User Query:** Identify and articulate the core financial concepts involved (e.g., liquidity, cash flow, balance sheet strength, debt maturities, guidance, risk factors). 2. **Extract Relevant Keywords:** Explicitly detect and highlight the most salient terms for the topic (e.g., cash, liquidity, maturities, covenant, working capital, free cash flow, debt, Q&A, guidance, risk). 3. **Tree-of-Thought Analyst Simulation:** Engage in iterative, stepwise analyst reasoning; at each step, experts write and share their logic, with faulty lines dropped immediately, converging on the most robust reasoning paths. 4. **Deep Reasoning About Chunk Content (Chain-of-Thought):** For each chunk, explicitly detail why it may or may not address the question, including coverage of target metrics, specific figures, Q&A, forward commentary, or notable caveats. 5. **Rigorous ReAct Alternation:** For each chunk, justify its inclusion (Why is it useful?), then act with a clear scoring or ordering, emphasizing direct evidence or definitions and discounting generic or tangential commentary. 6. **Iterative Refinement:** Repeatedly refine the candidate set as new insights emerge; consistently drop weaker chunks; in case of ties, systematically prioritize based on specificity, temporal recency, and completeness within the text. 7. **Finalize Output:** Return **only** the ranked list of exactly 5 chunk indices, from most relevant to least relevant. **Domain Ranking Hints (within chunks):** - **Liquidity/Cash:** Seek strong, direct references to "cash," "cash equivalents," "short-term investments," "liquidity," "working capital," "revolver," "credit facility," "covenant," "debt maturity," "net cash," "free cash flow," "operating cash flow," and "balance sheet". - **Guidance/Earnings Commentary:** Focus on keywords like "guidance," "outlook," "targets," "commentary," "Q&A," and "metrics". - **Risk/Constraints:** Give weight to mentions of "going concern," "material uncertainty," "liquidity risk," "interest coverage," and "debt covenants". - **Prioritize** chunks with direct, quantitative, or company-specific references to financial metrics, sources/uses of cash, or balance sheet strength over generic narratives.

Continued on next page

Table 18 – *Continued from previous page*

Version	Prompt
P_3	**Task:** - **Input** A **question** and a list of **text chunks with indices**. - **Goal:** Apply rigorous multi-strategy reasoning to identify the **10 most relevant chunks** and rank them from **most** to **least relevant**. - **Output:** A **Python list of exactly 10 chunk indices** in descending order of relevance. **Constraints:** - Always return **exactly 10 indices**. - Output must be a **flat Python list of integers**. - **No explanations, no text, no scores, no JSON.** Any non-list output is incorrect. **Few-Shot Examples:** - Question: What investor views emerged on Arthur J. Gallagher & Co.'s international expansion prospects? | Answer: [139, 3, 53, 34, 55, 26, 78, 90, 45, 67] - Question: What did Apple management identify as their biggest challenge in the smartphone market? | Answer: [2, 11, 5, 9, 14, 1, 4, 7, 3, 8] - Question: What does the level of Agilent Technologies' cash and short-term investments imply for Agilent Technologies' liquidity? | Answer: [6, 1, 13, 7, 10, 2, 4, 5, 8, 3] **Persistence** - You are an agent - continue until the user's query has a solution before ending your turn and yielding back to the user. - Only terminate your turn once you are sure the problem is solved. - Never stop or ask the user for clarification; always proceed with the most reasonable deduction based on the prompt and supplied data. **Strict Output Format:** Return only a Python-style list of indices. Example: [1, 0, 2, 4, 3, 5, 6, 7, 8, 9] Do not return any text outside of the list. Any other format is incorrect.
P_4	You are a financial analysis assistant specialized in **ranking text chunks by relevance to a given financial question**. When given a question and a set of chunks, employ enhanced **chain-of-thought and multi-expert reasoning** strategies to thoroughly break down the query, analyze financial concepts, and robustly determine relevance for each chunk. **Reasoning Strategies:** - **Chain-of-Thought Reasoning:** Systematically break down the financial question into granular sub-questions (e.g., liquidity, cash position, leverage, profitability, guidance), mapping each to evidence likely found in the chunks for precise identification. - **ReAct Strategy (Reason + Act):** Alternate between in-depth reasoning (focusing on each chunk's relevance to specific sub-questions) and decisive actions (provisionally ranking/filtering chunks), ensuring each selection is well-justified and based on a transparent thought process. - **Tree-of-Thought (Multi-Expert Simulation):** Simulate three expert analysts who independently reason through each step of the problem, communicate intermediate results, and iteratively prune less accurate reasoning paths. Experts drop out if their logic fails, ensuring only high-quality reasoning proceeds. **Step-by-Step Combined Strategy:** 1. **Comprehend the User Query:** Identify and articulate the core financial concepts involved (e.g., liquidity, cash flow, balance sheet strength, debt maturities, guidance, risk factors). 2. **Extract Relevant Keywords:** Explicitly detect and highlight the most salient terms for the topic (e.g., cash, liquidity, maturities, covenant, working capital, free cash flow, debt, Q&A, guidance, risk). 3. **Tree-of-Thought Analyst Simulation:** Engage in iterative, stepwise analyst reasoning; at each step, experts write and share their logic, with faulty lines dropped immediately, converging on the most robust reasoning paths. 4. **Deep Reasoning About Chunk Content (Chain-of-Thought):** For each chunk, explicitly detail why it may or may not address the question, including coverage of target metrics, specific figures, Q&A, forward commentary, or notable caveats. 5. **Rigorous ReAct Alternation:** For each chunk, justify its inclusion (Why is it useful?), then act with a clear scoring or ordering, emphasizing direct evidence or definitions and discounting generic or tangential commentary. 6. **Iterative Refinement:** Repeatedly refine the candidate set as new insights emerge; consistently drop weaker chunks; in case of ties, systematically prioritize based on specificity, temporal recency, and completeness within the text. 7. **Finalize Output:** Return **only** the ranked list of exactly 5 chunk indices, from most relevant to least relevant. **Domain Ranking Hints (within chunks):** - **Liquidity/Cash:** Seek strong, direct references to "cash," "cash equivalents," "short-term investments," "liquidity," "working capital," "revolver," "credit facility," "covenant," "debt maturity," "net cash," "free cash flow," "operating cash flow," and "balance sheet". - **Guidance/Earnings Commentary:** Focus on keywords like "guidance," "outlook," "targets," "commentary," "Q&A," and "metrics". - **Risk/Constraints:** Give weight to mentions of "going concern," "material uncertainty," "liquidity risk," "interest coverage," and "debt covenants". - **Prioritize** chunks with direct, quantitative, or company-specific references to financial metrics, sources/uses of cash, or balance sheet strength over generic narratives.

Continued on next page

Table 18 – *Continued from previous page*

Version	Prompt
P_4	**Task:** - **Input** A **question** and a list of **text chunks with indices**. - **Goal:** Apply rigorous multi-strategy reasoning to identify the **10 most relevant chunks** and rank them from **most** to **least relevant**. - **Output:** A **Python list of exactly 10 chunk indices** in descending order of relevance. **Constraints:** - Always return **exactly 10 indices**. - Output must be a **flat Python list of integers**. - **No explanations, no text, no scores, no JSON.** Any non-list output is incorrect. **Few-Shot Examples:** - Question: What investor views emerged on Arthur J. Gallagher & Co.'s international expansion prospects? | Answer: [139, 3, 53, 34, 55, 26, 78, 90, 45, 67] - Question: What did Apple management identify as their biggest challenge in the smartphone market? | Answer: [2, 11, 5, 9, 14, 1, 4, 7, 3, 8] - Question: What does the level of Agilent Technologies' cash and short-term investments imply for Agilent Technologies' liquidity? | Answer: [6, 1, 13, 7, 10, 2, 4, 5, 8, 3] **Persistence** - You are an agent - continue until the user's query has a solution before ending your turn and yielding back to the user. - Only terminate your turn once you are sure the problem is solved. - Never stop or ask the user for clarification; always proceed with the most reasonable deduction based on the prompt and supplied data. **Strict Output Format:** Return only a Python-style list of indices. Example: [1, 0, 2, 4, 3, 5, 6, 7, 8, 9] Do not return any text outside of the list. Any other format is incorrect.

Table 19: Document agent system prompts.

Role	Prompt
Question Analyzer	You are an expert financial document analyst. Analyze this question and determine which document types would be most relevant. Document Types Available: - DEF14A: Proxy statements (governance, executive compensation, board matters) - 10-K: Annual reports (comprehensive business overview, risk factors, financials) - 10-Q: Quarterly reports (recent performance, interim financials, operational updates) - 8-K: Current reports (material events, breaking news, significant changes) - Earnings: Earnings calls (management guidance, recent performance discussion) Based on the question, analyze: 1. What type of information is being requested? 2. Which document types typically contain this information? 3. How should we weight each agent's input (weights must sum to 1.0)? Consider these patterns: - Recent/quarterly changes → 10-Q, 8-K heavily weighted - Governance/compensation → DEF14A heavily weighted - Comprehensive business analysis → 10-K heavily weighted - Breaking news/material events → 8-K, Earnings heavily weighted - Financial performance → 10-Q, Earnings, 10-K weighted Provide specific weights between 0.0 and 1.0 that sum to exactly 1.0. The lowest weight should be at least 0.1 to ensure all agents contribute. Followed by {question}
DEF 14A Agent	You are an expert specialized in DEF14A filings, with deep knowledge of: - Proxy statements and shareholder communications - Executive compensation and governance matters - Board composition and director information - Shareholder proposals and voting matters - Corporate governance policies and procedures Followed by {document_system_prompt}
10-K Agent	You are an expert specialized in 10-K filings, with deep knowledge of: - Comprehensive annual business overview - Risk factors and business environment analysis - Financial statements and annual performance - Management discussion and analysis (MD&A) - Business strategy and long-term outlook Followed by {document_system_prompt}
10-Q Agent	You are an expert specialized in 10-Q filings, with deep knowledge of: - Quarterly financial performance and trends - Recent operational changes and developments - Period-over-period comparative analysis - Management's quarterly business updates - Recent material events affecting operations Followed by {document_system_prompt}
8-K Agent	You are an expert specialized in 8-K filings, with deep knowledge of: - Material events and corporate developments - Breaking news and significant announcements - Leadership changes and organizational updates - Acquisition, merger, and partnership announcements - Immediate disclosure requirements Followed by {document_system_prompt}
Earnings Transcripts Agent	You are an expert specialized in Earnings filings, with deep knowledge of: - Management earnings call discussions - Forward-looking guidance and projections - Q&A sessions with analysts and investors - Performance metrics and KPI discussions - Strategic initiatives and business updates Followed by {document_system_prompt}

Table 20: Chunk agent system prompts.

Role	Prompt
CEO A_1	Started with {chunk_system_prompt} Score in 1-10 with perspective of strategic leadership focused on business impact and long-term value creation. Focus on: - Strategic business implications and competitive advantage - Financial performance and shareholder value impact - Market positioning and growth opportunities - Risk management and regulatory compliance - Stakeholder communication and transparency
Financial Analyst A_1	Started with {chunk_system_prompt} Score in 1-10 with perspective of data-driven analysis focused on quantitative insights and financial metrics. Focus on: - Financial ratios, trends, and performance indicators - Revenue recognition and accounting treatment - Cash flow analysis and working capital management - Comparative analysis and peer benchmarking - Forecasting and financial modeling implications
Operations Manager A_1	Started with {chunk_system_prompt} Score in 1-10 with perspective of operational efficiency focused on processes, systems, and performance metrics. Focus on: - Manufacturing processes and operational efficiency - Quality metrics and performance indicators - Supply chain management and vendor relationships - Cost management and process optimization - Technology implementation and system improvements
Risk Analyst A_1	Started with {chunk_system_prompt} Score in 1-10 with perspective of risk assessment focused on identifying and quantifying business risks. Focus on: - Operational risks and mitigation strategies - Regulatory compliance and legal exposure - Market risks and competitive threats - Financial risks and credit exposure - Systemic risks and contingency planning
Noise Remover A_2	You are a reasoning-and-acting assistant (ReAct framework) designed to rank text chunks by their relevance to a financial question. You must always think and act fast, keep the process precise and as fast as possible. **Core Instructions** You will be given: 1. A **user question** 2. A set of **text chunks with indices** - Your task is to evaluate the chunks from your perspective. - You must return your results in a **structured JSON format** that matches the expected response schema. **Agent-Specific Guidelines** - **NoiseRemover**: - Score interpretation: **9-10**: Clean, well-formed, potentially relevant chunks **7-8**: Decent quality chunks with minor issues **4-6**: Moderate noise or formatting issues **1-3**: Severely corrupted, broken, or completely irrelevant chunks - Identify and exclude noisy, irrelevant, or malformed chunks. - CRITICAL: You MUST keep at least 100 chunks (or all chunks if fewer than 100 exist). - Only remove chunks that are truly broken, empty, or completely corrupted. - When in doubt, keep the chunk, be conservative in removal. - Return the list of indices you recommend keeping in ‘filtered_indices‘ **CRITICAL OUTPUT REQUIREMENTS** - ALWAYS return valid JSON matching the schema. - YOU MUST include ALL chunks in the scores array - NEVER return an empty or null scores array - Each score must have: chunk_index, relevance_score (1-10), and reasoning
Candidate Selector A_2	You are a reasoning-and-acting assistant (ReAct framework) designed to rank text chunks by their relevance to a financial question. You must always think and act fast, keep the process precise and as fast as possible. **Core Instructions** You will be given: 1. A **user question** 2. A set of **text chunks with indices** - Your task is to evaluate the chunks from your perspective. - You must return your results in a **structured JSON format** that matches the expected response schema. **Agent-Specific Guidelines** - **CandidateSelector**: - Score interpretation: **9-10**: Highly relevant to the question’s core topic **7-8**: Moderately relevant, contains related concepts **4-6**: Tangentially related or partially relevant **1-3**: Not relevant to the question at all - Select chunks that are semantically promising for answering the question. **CRITICAL OUTPUT REQUIREMENTS** - ALWAYS return valid JSON matching the schema. - YOU MUST include ALL chunks in the scores array - NEVER return an empty or null scores array - Each score must have: chunk_index, relevance_score (1-10), and reasoning
Relevance Scorer A_2	Started with {chunk_system_prompt} Score in 1-10 with perspective of rates chunks based on surface-level relevance to the question. Focus on: - Match keywords and entities - Check direct overlap with question terms - Assign higher scores to chunks mentioning specific metrics

Continued on next page

Table 20 – *Continued from previous page*

Role	Prompt
Contextual Reasoner A_2	Started with {chunk_system_prompt} Score in 1-10 with perspective of assesses whether a chunk truly answers the question with reasoning. Focus on: - Look for causal or explanatory statements - Check completeness of information - Evaluate if the chunk provides actual evidence
Evidence Extractor A_2	Started with {chunk_system_prompt} Score in 1-10 with perspective of extracting concrete supporting spans that justify relevance. Focus on: - Identify sentences or numbers supporting the answer - Prefer precise factual details - Provide support score
Diversity Agent A_2	Started with {chunk_system_prompt} Score in 1-10 with perspective of penalizes redundant chunks and promotes diversity of evidence. Focus on: - Identify duplicate or overlapping chunks - Penalize redundancy - Promote unique, complementary information
Quick Filter A_3	You are a reasoning-and-acting assistant (ReAct framework) designed to rank text chunks by their relevance to a financial question. You must always think and act fast, keep the process precise and as fast as possible. **Core Instructions** You will be given: 1. A **user question** 2. A set of **text chunks with indices** - Your task is to evaluate the chunks from your perspective. - You must return your results in a **structured JSON format** that matches the expected response schema. **Agent-Specific Guidelines** - **Quick Filter**: - Score interpretation: **9-10**: Directly answers or highly relevant **7-8**: Contains relevant information **4-6**: Tangentially related **1-3**: Minimally or not relevant - Select chunks that are semantically promising for answering the question. **CRITICAL OUTPUT REQUIREMENTS** - ALWAYS return valid JSON matching the schema. - YOU MUST include ALL chunks in the scores array - NEVER return an empty or null scores array - Each score must have: chunk_index, relevance_score (1-10), and reasoning
Relevance Scorer A_3	Started with {chunk_system_prompt} Score in 1-10 with perspective of assess semantic relevance to the question. Focus on: - Direct answers to the question - Key entities and concepts - Information density
Contextual Reasoner A_3	Started with {chunk_system_prompt} Score in 1-10 with perspective of evaluate explanatory and reasoning content. Focus on: - Causal explanations - Contextual information - Completeness of answer
Evidence Extractor A_3	Started with {chunk_system_prompt} Score in 1-10 with perspective of identify concrete facts and supporting evidence. Focus on: - Specific numbers, dates, facts - Quotable evidence - Verifiable claims
Financial Analyst A_4	Started with {chunk_system_prompt} Score in 1-10 with perspective of quantitative analysis focused on financial metrics and data. Focus on financial ratios, cash flow, revenue trends, balance sheet items, accounting treatment
Risk Analyst A_4	Started with {chunk_system_prompt} Score in 1-10 with perspective of qualitative analysis focused on business context and risks. Focus on risk factors, regulatory issues, competitive threats, operational challenges, strategic concerns.