

Parallel and Multi-Stage Knowledge Graph Retrieval for Behaviorally Aligned Financial Asset Recommendations

Fernando Spadea^{1,*}, Oshani Seneviratne¹

¹Rensselaer Polytechnic Institute, Troy, NY, USA

Abstract

Large language models (LLMs) show promise for personalized financial recommendations but are hampered by context limits, hallucinations, and a lack of behavioral grounding. Our prior work, FLARKO, embedded structured knowledge graphs (KGs) in LLM prompts to align advice with user behavior and market data. This paper introduces RAG-FLARKO, a retrieval-augmented extension to FLARKO, that overcomes scalability and relevance challenges using multi-stage and parallel KG retrieval processes. Our method first retrieves behaviorally relevant entities from a user's transaction KG and then uses this context to filter temporally consistent signals from a market KG, constructing a compact, grounded subgraph for the LLM. This pipeline reduces context overhead and sharpens the model's focus on relevant information. Empirical evaluation on a real-world financial transaction dataset demonstrates that RAG-FLARKO significantly enhances recommendation quality. Notably, our framework enables smaller, more efficient models to achieve high performance in both profitability and behavioral alignment, presenting a viable path for deploying grounded financial AI in resource-constrained environments.

Keywords

Retrieval Augmented Generation, Knowledge Graphs, Large Language Model, Personalized Financial Recommendation, Behavioral Alignment, Multi-Stage Retrieval, Subgraph Extraction

1. Introduction

The task of **financial asset recommendation**, i.e., suggesting assets such as stocks or bonds to investors, is a knowledge-intensive process that requires personalization, transparency, and factual accuracy. While Large Language Models (LLMs) are powerful tools for generating financial advice, they face key limitations, including limited context windows and susceptibility to hallucinations. A critical challenge is their weak **behavioral grounding**: ensuring that recommendations align with an investor's past behavior and preferences, as reflected in their transaction history. An emerging question is whether generative AI can provide trusted, customized financial advice. Central to this is the need to ground generated advice in verifiable evidence, including market data or historical trends, rather than merely plausible-sounding text. Retrieval-augmented generation (RAG) offers a promising solution by enabling models to retrieve and incorporate real-time, external information into the recommendation process.

To address these challenges, our previous work on FLARKO (Financial Language-model for Asset Recommendation with Knowledge-graph Optimization) [1], introduced a framework that grounds LLM reasoning using two complementary knowledge graphs: a personal KG (PKG), capturing user transaction histories, and a market KG (MKG), representing market-level asset performance. The model is fine-tuned with Kahneman-Tversky Optimization [2] due to its performance under both centralized and federated settings [3]. The project's GitHub repository [1] contains the full documentation for the entire framework, including its centralized and federated implementations (CenFLARKO and FedFLARKO), as well as extensive evaluations against a range of baseline models for financial asset recommendations.

RAGE-KG 2025: The Second International Workshop on Retrieval-Augmented Generation Enabled by Knowledge Graphs, co-located with ISWC 2025, November 2–6, 2025, Nara, Japan

*Corresponding author.

✉ spadef@rpi.edu (F. Spadea); senevo@rpi.edu (O. Seneviratne)

🌐 <https://oshani.info> (O. Seneviratne)

🆔 0009-0006-4278-3666 (F. Spadea); 0000-0001-8518-917X (O. Seneviratne)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

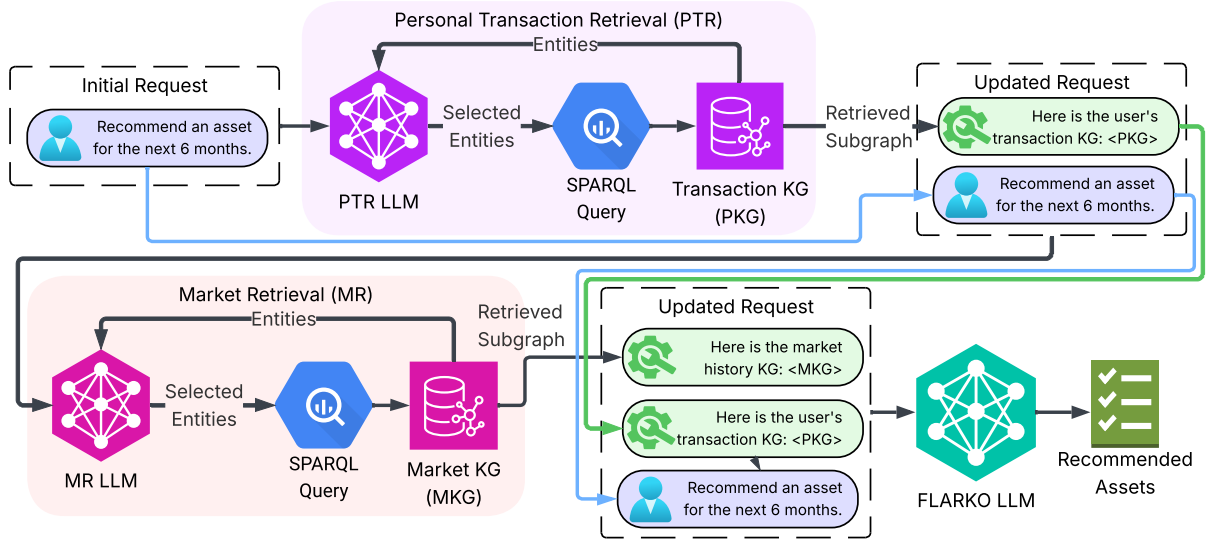


Figure 1: RAG-FLARKO Multi-Stage Retrieval Pipeline

While effective, the original FLARKO approach of full KG injection into the LLM context incurs significant token costs and struggles to scale to larger KGs or more complex queries. To address these limitations, in this paper, we introduce **RAG-FLARKO**, an extension of FLARKO that combines retrieval-augmented generation with structured KG reasoning. This approach enables more efficient “needle-in-a-haystack” retrieval in financial recommendation tasks by selectively injecting only the most relevant context. By decomposing KG inclusion into two targeted retrieval steps, RAG-FLARKO assembles a minimal, temporally filtered subgraph tailored to each user request. This design maintains behavioral grounding while addressing the scalability and context limitations of traditional KG injection directly into the LLM context. As shown in Figure 1, first, a *Personal Transaction Retrieval* (PTR) stage issues creates a relevant subgraph from a KG of the user’s transaction history. Next, a *Market Retrieval* (MR) stage creates a second relevant subgraph from a KG of the overall market, using the contextual information retrieved in the previous PTR stage. These subgraphs are then provided to the FLARKO model for responding to the user’s request. By decomposing retrieval with temporal filtering and leveraging KG structure, our pipeline overcomes LLM context limits and prevents data leakage, ensuring that all recommendations are grounded in facts available at query time.

Our Contributions:

- **Multi-step, KG-driven retrieval:** A two-stage RAG pipeline that issues SPARQL queries over user transaction and market KGs to chain behavior to relevant financial signals.
- **LLM-based entity selection and subgraph construction:** We apply LLMs to filter relevant entities and construct compact, context-efficient KG subgraphs for recommendation generation.
- **Context-window optimization via RAG-FLARKO:** Our method reduces token footprint, enabling smaller LLMs to outperform full-KG models in behavioral alignment and profitability.
- **Empirical evaluation in financial recommendation:** We evaluate RAG-FLARKO on the FAR-Trans dataset [4], demonstrating improved recommendation quality over baseline FLARKO in terms of both profitability and behavioral alignment.

2. Related Work

2.1. KGs in Financial Recommendation Systems

KGs have become a core tool for enhancing recommender systems, providing relational structure and semantic context that go beyond user–item interaction matrices [5]. In financial applications, where relationships between investors, instruments, markets, and institutions are intricate and dynamic, KGs enable more explainable and adaptable recommendations [6]. In consumer-facing applications such as robo-advisors, KGs help link investor preferences with product attributes to produce tailored suggestions [7]. In financial news recommendation, KGs help encode entities and events to contextualize article relevance based on user portfolios or interests [8]. This illustrates a broader trend: KGs enhance personalization and interpretability by embedding domain knowledge into the recommendation pipeline.

Several systems further demonstrate the utility of KGs:

- FinDKG [9] constructs a dynamic knowledge graph from real-time financial news, capturing evolving sector trends for thematic investment strategies. While powerful for macroeconomic reasoning, FinDKG is not tailored to individual user behavior scenarios like ours.
- Tang et al. [10] present a stock recommender based on a generalized financial KG representing companies, industries, and market signals. Their model supports thematic matching via graph traversal but does not support behavior-specific filtering or LLM-based recommendation generation.
- Verma et al. [11] introduce an interpretable article recommender that builds a KG from structured and unstructured data, supporting both XGBoost and reinforcement learning-based inference. While they highlight the value of KG traversal paths, their system focuses on content recommendation rather than asset allocation or behavioral alignment.
- The FNRKPL framework [6] translates KG triples into natural language prompts to guide news recommendation. This approach injects KG-derived facts into LLM prompts but lacks structured subgraph retrieval and personalization, two core features of our RAG-FLARKO pipeline.

In contrast to these works, our method targets behaviorally aligned financial asset recommendation using structured personal transaction and market KGs, and introduces a multi-stage retrieval process to construct compact, temporally valid, user-specific KG subgraphs.

2.2. RAG for Financial Tasks

Originally proposed by Lewis et al. [12], RAG combines a retriever with a generative language model, enabling outputs that are grounded in external information. RAG has proven effective in factual domains such as law and medicine, and is increasingly being adopted for financial tasks. In finance, RAG has been explored to improve factual correctness and reduce hallucination:

- Shah et al. [13] introduce KG_RAG and RAG_SEM, two architectures for multi-document financial QA over earnings reports and filings. KG_RAG incorporates retrieved triples into LLM input, but it assumes a document-centric setup and lacks a mechanism for behavioral personalization or temporal integrity.
- FinSRAG [14] adapts RAG to time series forecasting by retrieving historical price patterns similar to current ones. These are then fed to a generation model (StockLLM) for commentary. While it shows the potential of RAG for quantitative forecasting, it does not use KGs or optimize for user preferences.
- SURGE [15], though not finance-specific, illustrates how KGs can support dialogue generation by grounding responses in structured knowledge. However, it does not perform multi-step or time-aware subgraph construction.

Unlike these approaches, our work integrates KG structure into RAG retrieval, using SPARQL-based multi-step filtering over both personal transaction and market KGs. We enforce temporal cutoffs

to ensure that no information leakage occurs, and we serialize the retrieved subgraphs in JSON-LD to preserve structure while fitting within LLM context limits. Most importantly, we align retrieval with user behavior (via PKG) and market signals (via MKG) to produce recommendations that are simultaneously personalized, profitable, and temporally grounded.

3. Methodology

Our framework extends the financial asset recommendation LLM framework, FLARKO [1], by employing a multi-step RAG pipeline that dynamically constructs a relevant knowledge base for submission to the LLM. This process ensures that the LLM’s recommendations are grounded in both personalized user behavior and timely market data, while remaining within the context limit of the LLM by prioritizing the most relevant information.

3.1. Knowledge Graph Design

While LLMs excel at processing and generating natural language, they require explicit contextual grounding for structured decision-making tasks to ensure interpretability, consistency, and robustness. To address this, our framework encodes financial context into two distinct KGs: a user’s Personal transaction KG (PKG) and a broader Market KG (MKG). These KGs serve as symbolic inputs that anchor the LLM’s reasoning to factual, structured data, mitigating common pitfalls like hallucination. By representing user behavior and market signals as interconnected triples serialized into JSON-LD, we provide the LLM with a transparent and controllable context [16], enabling it to reason over complex financial relationships to generate accurate and relevant recommendations.

- **Personal transaction KG (PKG):** This graph encodes an investor’s historical transaction behavior, serving as a proxy for their preferences, risk tolerance, and investment patterns. To construct a user’s transaction KG, we extract key features from their transaction logs, including the asset’s International Securities Identification Number (ISIN), the transaction type (buy/sell), its value, and its timestamp.
- **Market KG (MKG):** This graph captures broader market-level signals and asset characteristics. To manage the volume of raw price data and fit within the LLM’s context window, the historical price series are aggregated into `TenWeekPriceSummary` entities. Each summary encapsulates an asset’s performance over a ten-week interval, detailing the period’s high, low, average, and end prices. The MKG also includes descriptive asset metadata, such as its category, sector, and industry.

3.2. Multi-Step RAG Pipeline

The core of our methodology is a sequential, two-stage retrieval pipeline (as illustrated in Figure 1) that builds compact, relevant subgraphs from the comprehensive KGs to serve as context for the final recommendation task.

1. **Personal Transaction Retrieval (PTR):** The process begins with an initial user request. A retrieval-focused LLM is provided with this request and the set of all transaction entities from the user’s PKG. This LLM selects a subset of entities deemed most behaviorally relevant, i.e., aligned with the user’s past transaction history. From this selection, a SPARQL CONSTRUCT query is programmatically generated using the SPARQL query template given below, where `NODE_LIST` is a space-delimited list of the selected entities. This query retrieves a subgraph containing all triples where the selected entities appear as either the subject or the object, as shown in Figure 2a. The resulting subgraph, representing the most salient aspects of the user’s transaction history, is serialized and prepended to the original request via a system prompt.

2. **Market Retrieval (MR):** The updated request, now containing the retrieved PKG subgraph, proceeds to the market retrieval stage. A retrieval LLM is presented with this enriched context and the set of TenWeekPriceSummary entities from the MKG. The LLM selects relevant market entities, and the same SPARQL query template shown below is used to retrieve the corresponding market subgraph, as shown in Figure 2b. This market-focused subgraph is also serialized and prepended to the request via another system prompt.
3. **Recommendation Generation:** The final fully contextualized prompt, containing both the retrieved PKG and Market KG subgraphs, is passed to the FLARKO LLM. Grounded in this tailored, dual-faceted context, the model generates the final list of asset recommendations.

SPARQL Query Template

```

CONSTRUCT { ?s ?p ?o }
WHERE {
  VALUES ?node { [NODE_LIST] }
  { ?node ?p ?o . BIND(?node as ?s) }
  UNION
  { ?s ?p ?node . BIND(?node as ?o) }
}

```

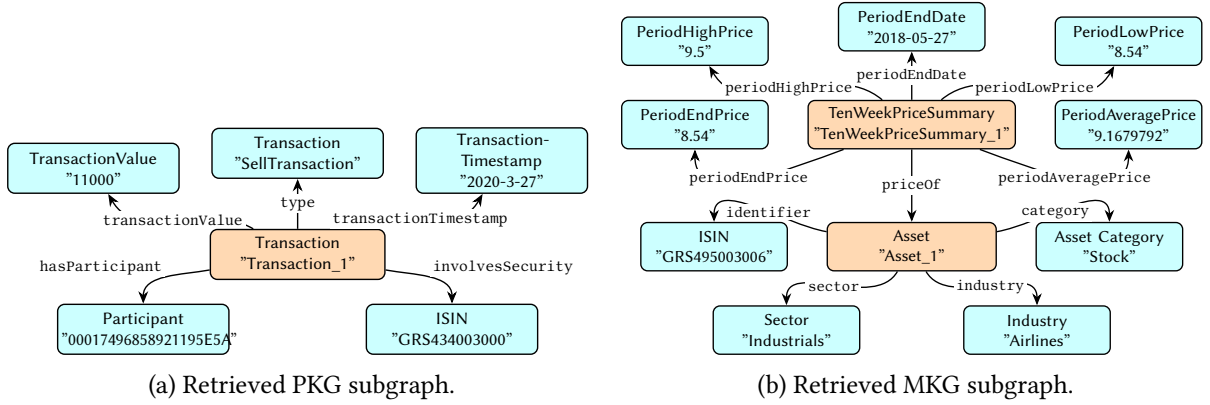


Figure 2: Examples of retrieved subgraphs from the (a) Personal transaction KG and (b) Market KG, each centered on a selected entity.

3.3. Dataset

We evaluate our framework using the **FAR-Trans dataset** [4], a real-world collection of anonymized financial data containing customer transaction histories, asset price data, and investor profile information. Our testing period spans from December 1, 2021, to November 29, 2022. Within this period, test instances are generated every two weeks, with each instance using the corresponding date as its RECOMMENDATION_DATE. This setup allows us to simulate a realistic scenario where recommendations are made periodically based on evolving historical data. A critical aspect of our approach is the enforcement of temporal consistency. For each recommendation request, the KGs are built using only information available before RECOMMENDATION_DATE. This strict cutoff prevents data leakage and ensures that the model’s recommendations are based on a historically accurate snapshot of information.

A concern with using pre-trained models like Qwen3 is the potential for data leakage, where the model may have been trained on the evaluation dataset. However, the pre-training process for Qwen3 involved sourcing data primarily from general web crawls, PDF-like documents, and synthetic data generation focused on domains such as mathematics and coding [17]. This methodology suggests it is unlikely that a specialized dataset like FAR-Trans was explicitly included in the training corpus. Furthermore, even in the event of some data overlap, the nature of our task, which grounds recommendations in the

specific user and market KGs provided in the prompt, significantly mitigates the impact of any potential data leakage.

4. Evaluation

4.1. Evaluation Metrics

To measure the quality of the recommendations, we use three variants of the **Hits@3** metric, which assesses the top three assets recommended by the model. Each metric evaluates the recommendations against outcomes in the 180-day period following the `RECOMMENDATION_DATE`.

- **Pref@3 (Preference Alignment):** This measures the hit rate of the recommendations against the set of assets that the user *actually purchased* during the subsequent 180-day window. It quantifies how well the model aligns with the user’s revealed preferences.
- **Prof@3 (Profitability):** This measures the hit rate against assets that yielded a *positive financial return* over the same 180-day period. It assesses the financial soundness of the recommendations.
- **Comb@3 (Combined Score):** This is our primary metric for success, as it measures the hit rate against the intersection of the two sets above, assets that were *both purchased by the user and were profitable*. A high Comb@3 score indicates that the model is generating actionable, high-quality advice that is both behaviorally aligned and financially beneficial.

4.2. Baselines and Model Variants

We compare the performance of RAG-FLARKO against:

1. **FLARKO Baseline:** The original FLARKO framework, which injects both the PKG and MKG directly into the LLM context without any intermediate retrieval steps. This baseline highlights the limitations of full KG injection in terms of context efficiency and scalability.
2. **Parallel RAG-FLARKO:** A baseline retrieval-augmented variant that applies the same two-stage pipeline but disables inter-stage context propagation. Specifically, the MR step operates without access to the context retrieved during the PTR stage. This isolates the impact of context chaining across retrieval stages.
3. **Multi-stage RAG-FLARKO:** Our full model, which performs sequential retrieval with context propagation from PTR to MR. This version enables the MR stage to incorporate behaviorally relevant information extracted from the PTR step, resulting in more targeted market signal retrieval and improved final recommendations.

We test all variants using the Qwen3-0.6B and Qwen3-1.7B LLMs [18] to highlight the benefits of context-efficient retrieval under limited model capacity.

5. Results

Overall Performance Gains: The empirical results, presented in Figure 3, demonstrate the significant advantages of the RAG-FLARKO framework over the baseline FLARKO models. For both the Qwen3-0.6B and Qwen3-1.7B models, the RAG-FLARKO (both parallel and multi-stage) implementations yield substantial improvements across almost all key metrics. Most notably, the right panel shows a dramatic increase in the **Comb@3** score, our primary metric for evaluating recommendations that are simultaneously profitable and aligned with user preferences.

This performance gain underscores the efficacy of our retrieval-augmented approach. The baseline FLARKO models, which fill the entire context with the KGs, struggled due to the limited context capacities of the smaller Qwen3 models. In contrast, RAG-FLARKO’s retrieval pipeline constructs compact and highly relevant subgraphs, enabling more efficient use of the available context window. This targeted approach allows the LLM to ground its reasoning in the most salient user and market data, leading to higher-quality recommendations.

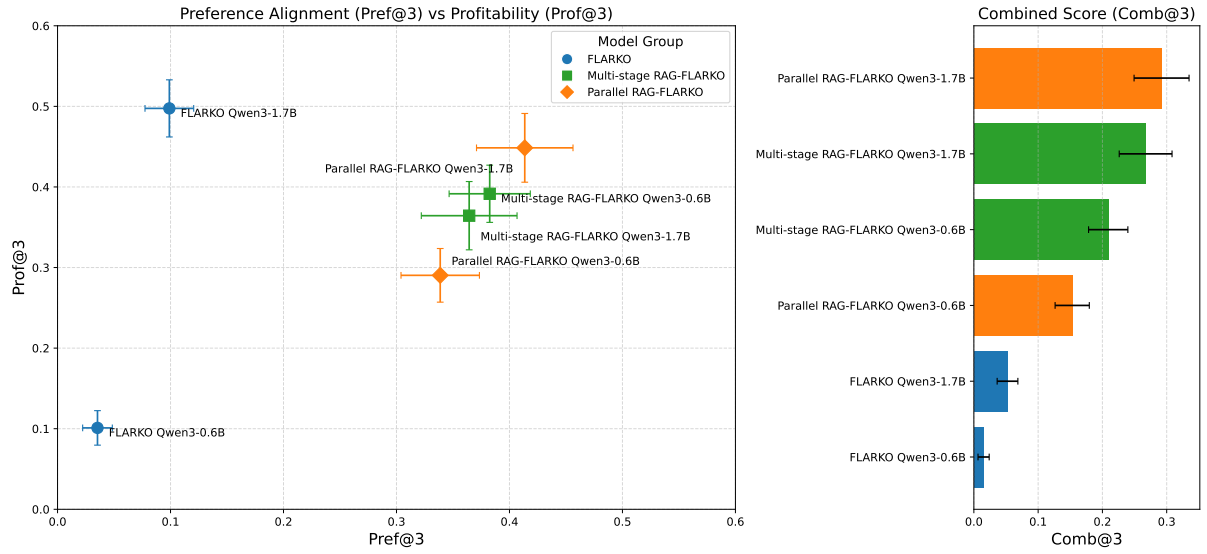


Figure 3: Performance Comparison of FLARKO, RAG-FLARKO, and Parallel RAG-FLARKO
The left panel plots preference alignment (Pref@3) against profitability (Prof@3) for all models. The right panel reports Comb@3, which quantifies how often the model recommends assets that are both profitable and behaviorally aligned. The Standard Error of Proportions are plotted in the graphs as well.

Impact of Inter-Stage Context: Our comparison between the multi-stage RAG-FLARKO pipeline and a parallel variant, confirms the value of contextual chaining between retrieval stages in the smaller Qwen3_0.6B. As shown in Figure 3, the multi-stage Qwen3_0.6B model consistently outperform its parallel counterpart.

The parallel RAG framework, which retrieves personal and market information independently, lacks the user-specific context during market signal selection. The multi-stage pipeline, however, uses the output of the PTR stage to inform the MR stage, which ensures that the retrieved market data from the MKG is conditioned on the user’s unique behavioral patterns in the PKG. The resulting improvement in all metrics highlights that this inter-stage context is crucial for identifying financial signals that are not just broadly relevant, but specifically tailored to the individual investor, thereby enhancing both profitability and preference alignment.

Impact of Model Size: Interestingly, the benefits of the inter-stage context do not apply to the larger Qwen3_1.7B model, which does not improve with the multi-stage pipeline. This suggests that larger models may possess sufficient reasoning capacity to handle entity selection without this additional context, whereas smaller models require that explicit guidance to perform optimally.

The observation that the multi-stage RAG pipeline disproportionately benefits the smaller Qwen3-0.6B model is a critical finding, supporting the feasibility of deploying RAG-FLARKO in real-world, resource-constrained environments. While the addition of inter-stage context did not significantly help the larger Qwen3-1.7B model, it transformed the Qwen3-0.6B model from the poorest performer into a highly competitive one. This result is particularly relevant to the deployment considerations discussed in Section 6, where smaller, quantized models are ideal for execution on edge servers or client devices to preserve privacy and reduce latency. The success of the enhanced smaller model demonstrates a practical pathway for creating effective financial recommendation systems that do not rely on large, resource-intensive models, thereby making advanced, personalized AI tools more scalable and accessible. This makes the multi-stage pipeline a valuable approach in federated learning environments, where smaller, more efficient models are often necessary due to resource constraints on client devices.

6. Discussion

Rationale for Multi-Step Retrieval: While our multi-step retrieval process incurs an additional LLM call over just having a single-step process (in which PKG and MKG are merged and queried jointly), this design yields important benefits. The single-step process introduces several limitations that negatively impact both retrieval quality and model interpretability. Most critically, it reduces the contextual information for the LLM to select its list of relevant entities. In contrast, the multi-step process enables the MR stage to condition its entity selection on behavioral context retrieved during the PTR stage. The context retrieved in the PTR stage includes a lot of important information from the user’s PKG that informs the LLM of the user’s specific preferences, so it can focus on the relevant entities. On the other hand, a single step would have the LLM select relevant entities for both the PKG and MKG without the previously retrieved context from within the PKG, since it would be working off only the list of entities. Additionally, it would also make the task much harder as the LLM would have to reason over the combined list of entities all at once rather than separately, overwhelming the LLM due to the PKG and MKG’s entities representing different types of information.

MKG Summaries: The market graph is massive, and fitting it in an LLM’s context window is infeasible. While our RAG framework alleviates this issue through selective retrieval, the use of TenWeekPriceSummary entities in the MKG further improves efficiency and interpretability. Individual daily data points, though granular, take up almost as much space as a summary in the model’s context window. With summaries, we can provide much more information, over a much longer time span, to the LLM than we can with daily data points. Additionally, it is much more effective to generalize patterns with macro data than with micro data.

Deployment Considerations: The modular architecture of the RAG-FLARKO pipeline lends itself well to deployment in resource-constrained or privacy-sensitive environments. The two retrieval stages (PTR and MR) operate over structured KGs and can be executed locally on client devices or edge servers. This design allows users to retain control over sensitive transaction data and minimizes the need to transmit personal information to centralized infrastructure.

While our current implementation uses the same LLM for all three stages, a natural future optimization would be to substitute a lightweight model for the PTR component. Because PTR focuses solely on selecting behaviorally relevant entities from the user’s PKG, it could be effectively handled by a smaller LLM that has been quantized for edge deployment. This would reduce memory and compute requirements, enabling real-time inference on mobile or embedded devices.

Only the final recommendation generation step requires access to a full-scale LLM, which can be hosted in a secure cloud environment or exposed via a privacy-preserving API. This separation of concerns enables hybrid deployment strategies that reduce latency, improve scalability, and uphold data sovereignty.

Emerging techniques such as model quantization and distillation can further enhance RAG-FLARKO’s deployability. Quantized models can dramatically reduce memory footprint and inference latency with minimal performance loss, making them ideal for on-device execution. Cross-device inference approaches could also be explored, where retrieval runs on the user’s device while generation is offloaded to a remote server, maintaining a balance between responsiveness and resource efficiency. For example, in mobile-based financial advisory systems, users could locally construct personalized subgraphs and perform retrieval in real time, sending only distilled, behavior- and market-relevant context to a centralized LLM.

7. Conclusion

RAG has emerged as a promising approach to tackle the knowledge-intensiveness of financial AI tasks. This is particularly important in finance, where pretrained models may lack access to up-to-date

information or personalized behavioral insights needed for trustworthy recommendations.

Our RAG-FLARKO framework introduces a multi-step retrieval pipeline designed to efficiently navigate and inject only the most relevant knowledge subgraphs into the LLM’s context. This architecture is optimized for compact models, addressing context limitations while maintaining personalization and factual grounding. We include empirical comparisons against the original FLARKO baseline to evaluate the overall improvement from introducing multi-step and parallel retrieval.

Our empirical evaluation validates this promise, demonstrating that the RAG-FLARKO framework delivers substantial gains in recommendation quality. The results show that our method successfully generates recommendations that are both profitable and aligned with user behavior, as evidenced by consistent improvement in Comb@3 scores. Crucially, our ablation study revealed that the benefits of the sequential, multi-stage retrieval process are most pronounced for smaller, more resource-constrained models. This confirms that a structured, context-aware retrieval pipeline can effectively overcome the inherent limitations of smaller LLMs, elevating their performance to be competitive with much larger models. This finding provides a clear pathway for developing financially-grounded, personalized AI systems that are not only effective but also efficient and practical for real-world deployment.

Future work could explore integrating symbolic reasoning modules to assist the reasoning tasks currently handled by the LLM. For example, symbolic reasoning engines could enforce domain-specific constraints (e.g., regulatory rules, portfolio diversification requirements) during subgraph construction, ensuring that generated recommendations are not only grounded but are also compliant. Additionally, we could leverage inferred or declared user characteristics, such as customer type (mass, premium, legal entity, professional), the investors’ risk level (conservative, moderate, aggressive), and their investment capacity (large-scale, medium-scale, small-scale) to further personalize both the retrieval and generation stages of the pipeline. For instance, a risk-averse investor might benefit from a PTR stage that prioritizes historically stable assets and an MR stage that emphasizes downside protection signals (e.g., volatility indexes, bond yields). Conversely, an aggressive investor’s prompts could steer the pipeline toward high-growth sectors, momentum signals, or emerging asset classes. This adaptive prompting would allow the LLM to tailor both its interpretation of user behavior and its generation of recommendations, effectively embedding a user’s financial persona into the decision-making process.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT, Gemini and Grammarly in order to rephrase some of the sentences and also to fix grammar and spelling issues. After using these tools and services, the authors reviewed and edited the content as needed and take full responsibility for the publication’s content.

8. Resource Contributions

All research artifacts, including code and documentation, are released under the MIT license. To support transparency and reproducibility, we maintain an open-source GitHub repository [1] containing all software artifacts. The specific version of the code used in this paper is available under the tag RAGE-KG_2025 at https://github.com/brains-group/FLARKO/releases/tag/RAGE-KG_2025.

References

- [1] F. Spadea, O. Seneviratne, FLARKO: Financial Language-model for Asset Recommendation with Knowledge-graph Optimization, <https://github.com/brains-group/FLARKO/tree/RAG>, 2025.
- [2] K. Ethayarajh, W. Xu, N. Muennighoff, D. Jurafsky, D. Kiela, Kto: Model alignment as prospect theoretic optimization, 2024.

- [3] F. Spadea, O. Seneviratne, Federated fine-tuning of large language models: Kahneman-tversky vs. direct preference optimization, in: *Companion Proceedings of the ACM on Web Conference 2025*, 2025, pp. 1757–1760.
- [4] J. Sanz-Cruzado, N. Droukas, R. McCreadie, Far-trans: An investment dataset for financial asset recommendation, *arXiv preprint arXiv:2407.08692* (2024).
- [5] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, Q. He, A survey on knowledge graph-based recommender systems, *IEEE Transactions on Knowledge and Data Engineering* 34 (2020) 3549–3568.
- [6] S. Sun, X. Pan, S. Qi, J. Gao, Knowledge enhanced prompt learning framework for financial news recommendation, *Pattern Recognition* 163 (2025) 111461.
- [7] Z. Shen, Z. Wang, J. Chew, K. Hu, Y. Wang, Artificial intelligence empowering robo-advisors: A data-driven wealth management model analysis, *International Journal of Management Science Research* 8 (2025) 1–12.
- [8] J. Ren, J. Long, Z. Xu, Financial news recommendation based on graph embeddings, *Decision Support Systems* 125 (2019) 113115.
- [9] X. V. Li, F. Sanna Passino, Findkg: Dynamic knowledge graphs with large language models for detecting global trends in financial markets, in: *Proceedings of the 5th ACM international conference on AI in finance*, 2024, pp. 573–581.
- [10] C. M. Tang, Y. Zhao, X. Yu, Intelligent stock recommendation system based on generalized financial knowledge graph, in: *Third International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI 2022)*, volume 12509, SPIE, 2023, pp. 332–338.
- [11] G. Verma, S. Sengupta, S. Simanta, H. Chen, J. A. Perge, D. Pillai, J. P. McCrae, P. Buitelaar, Empowering recommender systems using automatically generated knowledge graphs and reinforcement learning, *arXiv preprint arXiv:2307.04996* (2023).
- [12] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, *Advances in neural information processing systems* 33 (2020) 9459–9474.
- [13] S. Shah, S. Ryali, R. Venkatesh, Multi-document financial question answering using llms, *CoRR* (2024).
- [14] M. Xiao, Z. Jiang, L. Qian, Z. Chen, Y. He, Y. Xu, Y. Jiang, D. Li, R.-L. Weng, M. Peng, et al., Retrieval-augmented large language models for financial time series forecasting, *arXiv preprint arXiv:2502.05878* (2025).
- [15] M. Kang, J. M. Kwak, J. Baek, S. J. Hwang, Knowledge graph-augmented language models for knowledge-grounded dialogue generation, *arXiv preprint arXiv:2305.18846* (2023).
- [16] F. Spadea, O. Seneviratne, Bursting the filter bubble with knowledge graph inversion, in: *Companion Publication of the 17th ACM Web Science Conference 2025*, 2025, pp. 39–43.
- [17] Qwen Team, Qwen3: Think deeper, act faster, 2025. URL: <https://qwenlm.github.io/blog/qwen3/>.
- [18] Hugging Face, Qwen3-0.6b, <https://huggingface.co/Qwen/Qwen3-0.6B>, 2024.