Dynamic Graph-Retrieval Augmented Generation for Reliable and Explainable Consumer Electronics Recommendation

Jae Sung Park, Daeki Kim*, Jiman Kim*, Jiyoon Yoon, Seongwoon Jung

[1]Creative Lab. (C-Lab), Samsung Electronics
129, Samsung-ro, Yeongtong-gu, Suwon-si, Republic of Korea
(jason79.park,dkhi.kim,jm337.kim,jyoon.yoon,david24.jung)@samsung.com

Abstract

Although Large Language Models (LLMs) have expanded the capabilities of recommender systems, they are hindered by inherent limitations, including a propensity for factual hallucination and reliance on outdated domain knowledge. These constraints pose significant challenges in contexts requiring high-fidelity recommendations, such as consumer electronics purchases where precise, current specifications are critical. To address these issues, this paper proposes a novel framework termed Dynamic Graph-Retrieval Augmented Generation (RAG), which integrates LLMs, knowledge graphs, and RAG techniques within a multi-agent architecture. The framework dynamically deciphers complex user purchase intents and prioritizes decision-critical factors through a modular communication protocol that enables cross-agent collaboration, a 'Specification Vector Index' that resolves semantic disparities between natural language queries and technical attributes, and a graph-based dynamic retrieval engine that facilitates fact-grounded reasoning. Empirical validation forms a pivotal contribution of this work, with rigorous experimental verification confirming the system's efficacy in minimizing hallucinations through structured knowledge grounding. Quantitative metrics demonstrate statistically significant improvements in recommendation accuracy, such as 22.7% increase in precision, and reliability, while traceable decision pathways enhance operational transparency. This research delivers a foundational architecture for a possible practical recommender systems, validated through real-world deployment scenarios and test dataset, and establishes a benchmark for empirically substantiated innovation in AI-driven recommendation frameworks. By bridging theoretical innovation and practical deployment, this study marks a critical advancement in the field, offering both a new methodology and concrete evidence of its real-world applicability.

Introduction

Recently, the emergence of conversational search services leveraging Large Language Models (LLMs) has introduced a new paradigm, enabling consumers to find desired products through natural language (Wang et al. 2025; Lewis et al. 2020; Panarin 2025). LLMs show exceptional potential for understanding complex user queries, overcoming the limitations of conventional recommenders. This study focuses on the consumer electronics domain, where purchasing decisions require a high degree of accurate information regarding technical specifications, product comparisons, user reviews, expert evaluations, and price-performance value. Our proposed system aims to discern the user's latent intent to dynamically prioritize and deliver this critical information.

However, these powerful capabilities present a significant challenge. LLMs are susceptible to 'hallucination,' generating plausible but factually incorrect information, and their knowledge is static, failing to reflect rapidly changing product information or prices (Wang et al. 2025; Panarin 2025). This is a significant drawback for technology products where precise specifications are crucial. To address these limitations, Retrieval-Augmented Generation (RAG) technology has emerged as a solution to enhance factuality by referencing external knowledge sources (Lewis et al. 2020; Gao et al. 2023). Yet, conventional RAG methods that rely on unstructured text are inefficient at leveraging the structural and relational knowledge of product specifications. Meanwhile, Knowledge Graphs (KGs) are highly effective for representing this structured data and enhancing explainability (Huang and Huang 2024; Kwon, Ahn, and Seo 2024; Wang et al. 2019), but they cannot respond directly to natural language queries.

To address these interconnected challenges, we propose Dynamic Graph-RAG (D-GraphRAG), a new framework that synergistically combines the capabilities of LLMs, KGs, and RAG. The D-GraphRAG methodology operates in three core stages. First, an LLM parses the user's natural language query into structured features and constraints. Second, a novel Feature Description Vector Index (FDVI) bridges the semantic gap between user language and technical specifications, allowing the LLM to infer key features and their contextual importance. Finally, a graph query is auto-generated to retrieve and rank products, yielding a fact-based and traceable recommendation. Th FDVI database systematically classifies and describes various performance characteristics of products, enabling users to easily understand the importance and measurement methods of specific features. Each entry is provided in both Korean and English, and related terms and categories are offered to identify connections between features. Additionally, it provides quantitative data

*These authors contributed equally.

through measured attributes, allowing for objective comparisons.

Key contributions of this study include hallucination suppression through a hybrid RAG approach where user intent is connected to fact-based knowledge. An explainable service is provided by rendering the entire recommendation process transparent and traceable through the inferencing process. "Dynamic" denotes the system's capability to prioritize and adapt data in real time based on user intent. Static limitations are overcome by the D-GraphRAG framework, which flexibly interprets intent and links structured knowledge to enable dynamic services.

Related Work

This research integrates three key research streams: LLMs, RAG, and knowledge graph-based recommender systems. LLMs have opened new horizons in recommendation, with recent surveys classifying them into Discriminative and Generative paradigms (Wu et al. 2024; Liu et al. 2024; Lopez-Avila and Du 2025; Huang and Huang 2024). Our D-GraphRAG framework aligns with the generative paradigm. However, it is distinguished from existing research by its ability to go beyond simple prompting to dynamically retrieve and reason with structured external knowledge, which also has the potential to mitigate the cold-start problem (Wang et al. 2024; Balloccu et al. 2024; Zhang et al. 2025). RAG has emerged as a key technology for enhancing the factual accuracy of LLMs and reflecting up-to-date information (Lewis et al. 2020; Gao et al. 2023; Kwon, Ahn, and Seo 2024). However, in highly structured domains like consumer electronics, standard RAG exhibits clear limitations as it struggles to reason about complex entity relationships (Wang et al. 2025; Panarin 2025). For the representation of such structured information, KGs are a highly effective tool for mitigating data sparsity and enhancing explainability (Huang and Huang 2024; Kwon, Ahn, and Seo 2024; Wang et al. 2019), but they cannot respond directly to natural language queries on their own (Tourani, Nazary, and Deldjoo 2025; Wang et al. 2019). Furthermore, the success of Graph Neural Network (GNN) models like NGCF and LightGCN has demonstrated that leveraging graph topology is key to improving recommendation performance, providing the theoretical background for our adoption of a KG (He et al. 2020; Qiu et al. 2025).

Against this backdrop, recent trends are converging on integrating these three technologies. Our work is distinct from several approaches. Unlike conversational systems like G-CRS (Wang et al. 2024; Balloccu et al. 2024), D-GraphRAG focuses on deeply interpreting single, complex queries. In contrast to methods that use LLMs to generate a product knowledge graph (PKG) (Zhang et al. 2025; Peng et al. 2025), D-GraphRAG utilizes a pre-constructed graph as a 'source of truth' for real-time queries. Finally, while it shares principles with graph-augmented RAG like K-RagRec (Wang et al. 2025; Panarin 2025), our primary contribution is the FDVI. The FDVI acts as a 'semantic translation layer' that seamlessly connects the user's colloquial language with the KG's technical terminology. In conclusion, our research proposes a new architectural pattern that delineates the role of each component to maximize synergy, thereby ensuring both reliability and explainability.

Multi-Agent System Architecture

To overcome the flexibility and scalability limitations of conventional monolithic architectures, the D-GraphRAG framework adopts a modular design based on a Model Context Protocol (MCP) as illustrated in Fig. 1. The MCP is a communication protocol foundational to the multi-agent architecture, enabling seamless interaction between diverse agents and data sources. It operates on a three-tier structure (Host, Client, Server) where all data exchange occurs via a standardized payload with distinct Context and Data fields. This microservice-like structure offers clear advantages in fault isolation and flexible scalability, as new AI agents can be added simply by developing and integrating a new MCP Server without modifying existing system code. The framework consists of three logical layers operating via MCP. The Service Layer is the top-level user interface for receiving queries and presenting recommendations. The Orchestration Layer serves as the central control unit, analyzing queries, distributing tasks to agents, and synthesizing the final response using modules like an LLM-based Query Analysis Agent. The Agent Layer is a collection of specialized, independent servers, including an agent for user query analysis, a Vector DB agent for semantic retrieval, and a Graph DB agent for logical reasoning.

The system's knowledge base is kept current by an automated Database Manager pipeline that manages the data lifecycle—collecting, refining, and updating the Vector and Graph DBs. This is essential for addressing the LLM's inherent 'knowledge cutoff' problem. A primary challenge is the semantic gap between user language and the Graph DB's technical terminology. To address this challenge, we introduce the FDVI, a pre-generated index of natural language descriptions for technical features and contexts. When a user query is input, the system first searches the FDVI to find semantically relevant concepts. These concepts are then used to guide the LLM in generating a precise and factually-grounded graph query, which minimizes hallucination. The FDVI is composed of three distinct vector indexes for product (TVs), features and attributes, and expert verdicts to ensure targeted and accurate information retrieval. The accuracy of this process is further enhanced by constructing FDVI documents separately for each product model and employing contextual chunking. Ultimately, this integration of a tree-structured Graph DB with these three natural language-based Vector Indexes allows the system to leverage an LLM's interpretive power to query the graph via Cypher, efficiently processing complex questions and providing users with accurate, contextually relevant information.

Knowledge Graph for Product Recommendation

Conventional recommender systems struggle to understand users' complex natural language queries, failing to grasp

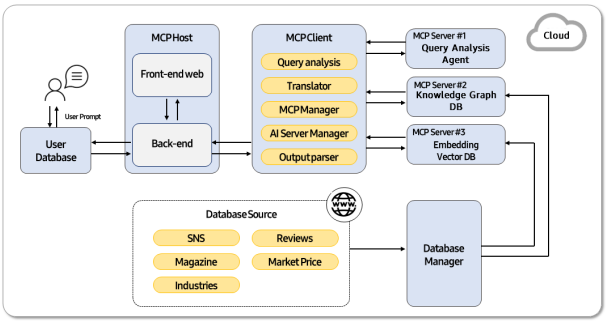

Figure 1: Architecture of a consumer electronics recommendation system composed of Multiple MCP servers (agents).

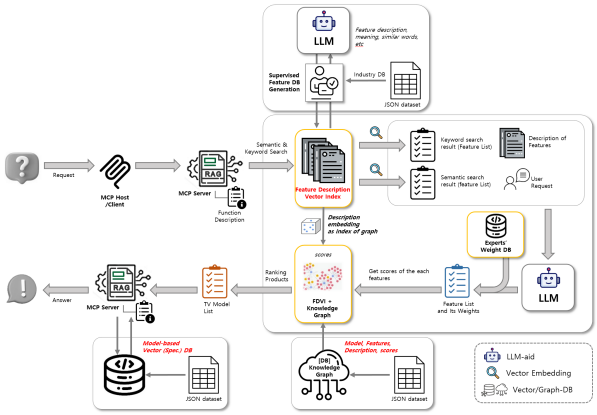

Figure 2: Flow of processing user prompts in the proposed framework.

the complex context in a request like, "a good TV for enjoying both movies and games in a bright living room". To overcome this, the D-GraphRAG methodology combines the reasoning of LLMs with the structured information of a Knowledge Graph. This approach aims for a paradigm shift by interpreting consumer queries to infer information priorities, performing dynamic personalization, and providing clear explanations for its recommendations.

The D-GraphRAG's multi-stage process can be formally interpreted as a systematic reduction of entropy—the measure of uncertainty about a user's true intent. An initial ambiguous query like 'a good TV' has very high entropy. The system's goal is to reduce the conditional entropy $H(X \mid Q)$ to near zero. Here, X represents the user's search intent embedded in the user query, and Q denotes the query itself, which is the user's input, respectively. The first stage (LLM parsing and FDVI search) performs a significant entropy reduction by resolving syntactic and semantic uncertainty, maximizing the mutual information between the query and the system's internal state. The second stage (Graph traversal) further reduces uncertainty through deterministic reasoning on the graph's structure. This two-stage process functions like a 'coarse-to-fine' search, efficiently reducing uncertainty by applying both hard and soft constraints to shrink the probability space of candidate products.

$$H(X) = -\sum_{i} P(x_i) \log_2 P(x_i) \tag{1}$$

A conventional vector DB RAG operates on unstructured text and document similarity, which is simpler to implement but suffers from a lack of relational understanding, poor explainability, and a higher potential for hallucinations. In contrast, the proposed 'KGs + Vector DB' RAG utilizes a structured, relationship-based approach. This enables more accurate, context-based query understanding, provides traceable explainability through graph nodes, and fundamentally minimizes hallucinations by grounding responses in existing data. While the graph-based method has a higher initial implementation complexity, its primary advantage is superior accuracy and reliability when handling complicated and relational data.

Prompt Preprocessing

As illustrated in Fig. 2, the initial stage converts a user's free-form natural language query into structured information. Our system utilizes an LLM as a JSON parser, which, with a single call, extracts multiple filtering conditions (e.g., size, brand) and keywords into a structured JSON object. This LLM-based parsing offers superior robustness and flexibility compared to traditional methods like regular expressions. An LLM's pre-trained Natural Language Understanding (NLU) capabilities allow it to accurately interpret varied and context-dependent expressions where conventional methods often fail (Lewis et al. 2020; Wu et al. 2024; Rajabi and Etminani 2024). This process can be viewed as 'Implicit Task Decomposition', where the LLM breaks down a single complex request into multiple independent constraints, sharing principles with the core Reasoning and Planning abilities of LLM agents (Lewis et al. 2020; Wang et al. 2024). For example, an user prompt "S brand, 65-inch, for living room use, bright TV" is parsed into a JSON object brand: 'S', size: '65', keywords: ['living room use', 'bright TV'], which forms the basis for the subsequent database query.

KG based Semantic Feature Selection and Weight Inferencing

After parsing the query, the system connects the user's intent with technical specifications and determines their importance. This core stage aims to map a subjective expression like 'bright living room' to an objective performance metric like 'high brightness' and dynamically determine its importance. To achieve this, we combine semantic search and LLM reasoning. Keywords and the full query sentence are embedded to search for candidate features in the prebuilt FDVI as illustrated in Fig. 3 (top). These candidates are then fed to the LLM to infer a final weight distribution. A key technical contribution here is using the FDVI to solve the 'vocabulary mismatch' problem, where general LLMs cannot connect user terms to specific technical features in a database (Huang and Huang 2024; Oh et al. 2024). The FDVI, an index of expert-written natural language descriptions for technical features, acts as a form of efficient

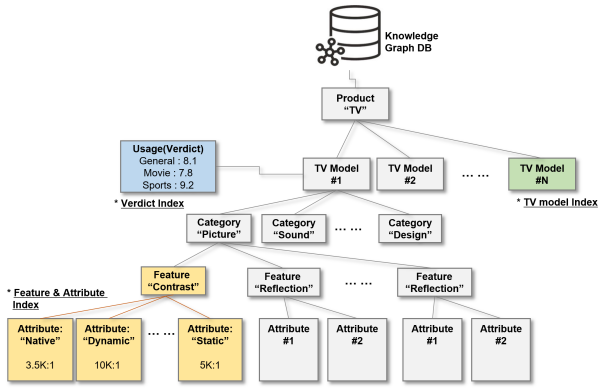

Figure 3: The knowledge graph structure for consumer electronics with complex specifications built in this study; details of this structure are described in experiemntal result section.

Knowledge Distillation, aligning with the core philosophy of RAG (Sun et al. 2019; Oh et al. 2024; Anuyah, Bolade, and Agbaakin 2024). Furthermore, the system's ability to infer feature weights in real-time based on query context enables dynamic personalization. A query for a 'cost-effective gaming TV' will prioritize performance and price, while one for a 'TV for a home with kids' might prioritize durability reviews. This on-the-fly optimization elevates the system from a simple search tool to an intelligent recommendation engine.

Cypher Query Generation and Product Recommendation

The final stage executes a database query to find the optimal product. The system automatically generates a Cypher query by combining the filters from stage 1 and the weighted features from stage 2 (Holzschuher and Peinl 2013). This query calculates a final personalized score for each product model using a weighted sum. This stage offers two key contributions. First, Explainability: The auto-generated Cypher query and the final JSON result provide a transparent audit trail of the recommendation process, addressing the 'black-box' problem. The output explicitly shows which filters and weighted features were used and includes an LLM-generated natural language explanation for the recommendation, an approach recognized as promising in recent explainable AI research (Balloccu et al. 2024; Zhang et al. 2025). Second, Structural result optimization: Leveraging the graph database's ability to handle relationships and traversals, the system ranks representative product (model) and then selects the single most suitable variant for each, which effectively prevents duplicate recommendations (Zhang et al. 2025; Yao et al. 2023). The final output is the top-ranked product, accompanied by a personalized explanation, such as, "This model was recommended because its 'brightness' and 'reflectance' performance are excellent, aligning with your stated need for a TV in a bright living room.".

Experimental Results

In this section, experiments were conducted to evaluate the performance of the proposed D-GraphRAG framework, particularly the core module that identifies the consumer's purchase intention. The experimental objectives are as follows. Identifying Core Information Requirements: Although the factors influencing consumer electronics purchasing decisions are complex, this study, based on prior research and market analysis, defines five key information types that consumers seek: (1) core technical specifications of the product, (2) detailed comparisons with alternative products, (3) hands-on reviews from similar user groups, (4) objective performance evaluation data from experts, and (5) information on retailers offering excellent price-performance value. Therefore, we set our first performance metric as: 'Does the proposed system accurately identify what information the consumer want to know from their natural language query?' Specifically, we quantitatively evaluate 'how accurately the proposed system infers the product specifications required by the user from the input of the consumer's unstructured natural language query' and 'how similar the specification-specific weights for product recommendation are to the weights assigned by experts'. To simplify this experiment, the dataset was built by limited to televisions (TVs), one of the most popular consumer electronics. Currently, the lack of a universal performance evaluation benchmark for consumer electronics recommendations based on specific product-related expert information results in the absence of a standardized method to assess the diverse and complex search and recommendation requests of consumers. Therefore, we constructed our own test bench dataset for the evaluation of this study. In the experimental, GPT-5-mini (OpenAI) was utilized as the primary LLM alongside the embedding model(text-embedding-3-large) to ensure a robust evaluation of the framework's performance.

Building User Prompt Dataset and Its Evaluation

A survey titled 'Consumer Electronics Purchase Prompt' was conducted to gather a test dataset of 300 natural language prompts simulating real-world product search scenarios. (See APPENDIX A: Users' prompt samples for TV purchases) To ensure dataset reliability, five electronics experts independently labeled the prompts, identifying when a user's intent required one or more of five key information types. Each prompt was classified using a 5-bit code, where each bit represents a distinct information type: basic specifications (MSB), competitor comparisons (4th digit), user reviews (3rd digit), expert reviews (2nd digit), and value-for-money analysis (LSB). A bit value of 1 (True) indicates the user requested the corresponding information. For example, a label of 11001 signifies requests for core technical specifications, detailed competitor comparisons, and value-for-money information. After consolidating expert opinions, 279 prompts with unanimous labeling results were selected as the final experimental dataset. The evaluated 5-digit codes for each user prompt was used to test the operational integrity of the proposed framework by detecting which of the five key information types holds the highest priority based on user prompts. The first stage of the framework,

user query analysis, determining whether the system should provide product-related information or tailored recommendations. Inputting 279 prompts into this step in the framework, its classification accuracy was measured by comparing inferred priority information (a multi-label output) with experts' labels as the ground truth. High accuracy validates the system's ability to understand user intent beyond simple keyword matching. Performance was assessed using the exact match ratio, where an 'exact match' occurs when the 5-digit code generated by the framework exactly matches the expert-annotated code. This strict metric evaluates performance conservatively, considering only perfect label matches as correct. The experiment achieved an inference accuracy of 96.74% in predicting expert-labeled data.

Our prompt analysis revealed that consumers primarily seek core technical specifications (requested in 90.7% of prompts) and value-for-money information ($\sim$82% of prompts), highlighting the importance of basic specifications and purchase utility. The most common prompt type, accounting for 41.2%, requested both 'technical specifications and value-for-money,' indicating a preference for concise core information. Requests for expert reviews (33% of prompts) reflected a desire for objective performance evaluations, while competitor comparisons ($\sim$23%) and user reviews ($\sim$21%) were less frequent. This study underscores the need for platforms to include core technical specifications and value-for-money information, alongside diverse data like expert evaluations to meet consumer needs.

Building Graph RAG Dataset and Its Evaluation

For our own KG for TVs, a variety of expert-level information, including product manufacturing specifications, marketing details, TV-related domain expertise, evaluations, and reviews, was primarily utilized. A knowledge graph was established for 53 representative TV products. The KG, termed TV-Graph, consists of 16,910 nodes connected by 16,822 relationships, with nodes organized into five distinct semantic categories. These include 12,349 Attribute nodes detailing technical specifications, 2,915 Feature nodes representing product functionalities, 636 Category nodes for grouping features by product type, 53 Product nodes corresponding to actual TV models, and 53 feature description nodes providing contextual explanations. Hierarchical relationships are structured such that 12,349 edges link feature nodes to their attribute nodes, 2,915 edges connect category nodes to feature nodes, and 636 edges associate product nodes with their respective category nodes. This architecture facilitates structured reasoning by establishing clear mappings between features, attributes, and product classifications.

Utilizing the constructed database, the second experiment evaluated the D-GraphRAG method's deeper capability for understanding and inferring user requirements by comparing it with the conventional RAG approach that uses only vector DB. This experiment was designed to evaluate the framework's ability to extract features and their weights which are used to deliver exact information or recommend products by deriving concrete technical specifications from a user's ambiguous expressions. In detail, this experiment was operateed as follows. To recommend the optimal product to the user, the proposed framework referenced and utilized an experts' product evaluation database. In this step, a crucial part of the framework was inferring the user's intended product usage purpose. Taking search information or recommendation of TV product as an example, the user's product "usage purpose" was subdivided into various items such as 'General use', 'Watching Movie', 'Play Gaming' etc., and the inference process determined which of these various items the user's prompt intended. In this process, with the assistance of the LLM, the prompt processor also considered semantic information from the user prompt. Specifically, for each potential "usage purpose", there is a feature list and weight vector in each product evaluation dataset which was predefined by an product experts group. We checked the similarity between the feature list and weight vector inferred by the proposed framework and the pre-defined by the expert group. Based on this result, we could measure how well the proposed framework could identify the user's product usage purpose and recommend products. High similarity signified that D-GraphRAG successfully inferred the consumer's subjective needs and extracted objective technical features of TV products, which was an essential prerequisite for reliable and personalized recommendations via LLM. Therefore, this experiment aimed to calculate and evaluate how similar inferred the feature list and estimated weight vector was to the judgment of the expert evaluation group as the ground truth. To quantitatively assess the performance of the proposed methods, we employed three standard information retrieval metrics: 'Precision', 'Recall', 'Normalized Discounted Cumulative Gain (NDCG)' and 'Jensen-Shannon Divergence (JSD)' (Zhang et al. 2016). Precision measured the accuracy of the retrieved features, indicating the proportion of recommended features that were relevant to the ground truth. Recall evaluated the completeness of the results, representing the fraction of all relevant features that were successfully identified by the method. Furthermore, NDCG was utilized to assess the quality of the ranking itself, assigning higher scores for placing more important features at the top of the list. JSD quantified the similarity of weight distributions between the proposed methods and the baseline. The test dataset comprised 17 user prompts (See APPENDIX A: Users' prompt samples for TV purchases), each with eight features ranked by assigned weights. Performance was measured using NDCG, precision, and recall for $k = 2$ to 8, with features selected in descending weight order. In the Table 1, features and their weight vectors are presented for the 17 user prompts for TV purchase.

Experimental results reveal critical insights into the performance of D-VectorRAG and D-GraphRAG relative to the conventional baseline (see Table 3). As k increased from 2 to 8, both methods exhibited a consistent decline in NDCG, precision, and recall, diverging from traditional information retrieval patterns where recall typically rises with k. This anomaly stems from the dynamic relevant set derived from the baseline's top-k features. Unlike fixed relevant sets in classic evaluations, the baseline's expanding feature pool at higher k introduced lower-weight features that D-VectorRAG and D-GraphRAG struggled to match. Consequently, the intersection between proposed and base-

D-VectorRAG																
	Feature #1	Feature #2	Feature #3	Feature #4	Feature #5	Feature #6	Feature #7	Feature #8	W1	W2	W3	W4	W5	W6	W7	W8
Prompt #1	Direct Reflections	Total Reflection	SDR Brightness	HDR Brightness	Black Level	Viewing Angle	HDR Color Volume	Color Saturation	0.25	0.2	0.2	0.12	0.1	0.07	0.04	0.02
Prompt #2	HDR Brightness	SDR Brightness	Black Level	Viewing Angle	HDR Color Volume	Color Saturation	SDR Color Volume	Unifrom Grayscale	0.2	0.2	0.18	0.14	0.12	0.08	0.05	0.03
Prompt #3	Direct Reflections	Total Reflection	Black Level	Viewing Angle	Resolutions	Unifrom Grayscale	Gaming @60Hz	Gaming @120Hz	0.22	0.18	0.14	0.13	0.12	0.1	0.06	0.05
Prompt #4	HDR Color Volume	Viewing Angle	Unifrom Blackness	Total Judder	Resolutions	Direct Reflections	Unifrom Grayscale	SDR Color Volume	0.2	0.18	0.17	0.13	0.12	0.12	0.04	0.04
Prompt #5	Resolutions	LQ Smoothing	Viewing Angle	Color Saturation	Unifrom Blackness	Build Quality	Frequency Response	VRR	0.22	0.22	0.18	0.12	0.12	0.06	0.05	0.03
Prompt #6	HDR Brightness	HDR Brightness	Total Judder	Resolutions	Direct Reflections	Viewing Angle	Frequency Response	LQ Smoothing	0.2	0.18	0.17	0.15	0.12	0.09	0.06	0.03
Prompt #7	Resolutions	Input Lag	VRR	Response Time	Total Judder	Gaming @60Hz	Gaming @120Hz	Frequency Response	0.25	0.22	0.18	0.12	0.1	0.07	0.05	0.01
Prompt #8	Contrast	Viewing Angle	Frequency Response	Direct Reflections	Black Level	SDR Brightness	Color Saturation	Response Time	0.28	0.22	0.15	0.12	0.1	0.06	0.05	0.02
Prompt #9	SDR Brightness	Direct Reflections	HDR Brightness	Viewing Angle	Black Level	Color Saturation	Build Quality	Resolutions	0.22	0.2	0.18	0.12	0.1	0.08	0.06	0.04
Prompt #10	Color Saturation	Unifrom Grayscale	Viewing Angle	Resolutions	Direct Reflections	Black Level	LQ Smoothing	VRR	0.26	0.2	0.16	0.16	0.11	0.06	0.04	0.01
Prompt #11	Unifrom Blackness	HDR Color Volume	Total Judder	Viewing Angle	Resolutions	Frequency Response	LQ Smoothing	Unifrom Grayscale	0.25	0.22	0.18	0.12	0.1	0.06	0.05	0.02
Prompt #12	HDR Color Volume	Resolutions	LQ Smoothing	Viewing Angle	Unifrom Grayscale	Total Judder	VRR	Frequency Response	0.28	0.23	0.18	0.1	0.08	0.06	0.04	0.03
Prompt #13	Unifrom Blackness	Total Judder	HDR Brightness	Resolutions	Viewing Angle	LQ Smoothing	Unifrom Grayscale	Build Quality	0.22	0.18	0.18	0.15	0.1	0.08	0.05	0.04
Prompt #14	Resolutions	SDR Brightness	Viewing Angle	Stutter	LQ Smoothing	Total Judder	Unifrom Blackness	VRR	0.2	0.18	0.17	0.15	0.12	0.09	0.06	0.03
Prompt #15	Viewing Angle	LQ Smoothing	Build Quality	Resolutions	SDR Color Volume	HDR Color Volume	Frequency Response	Total Judder	0.27	0.22	0.18	0.12	0.11	0.06	0.03	0.01
Prompt #16	Viewing Angle	HDR Color Volume	SDR Color Volume	Build Quality	SDR Brightness	Frequency Response	Total Judder	Resolutions	0.25	0.15	0.15	0.12	0.12	0.08	0.08	0.05
Prompt #17	LQ Smoothing	Resolutions	Direct Reflections	Black Level	Color Saturation	Viewing Angle	Frequency Response	Unifrom Blackness	0.27	0.22	0.18	0.16	0.08	0.05	0.02	0.02
D-GraphRAG																
	Feature #1	Feature #2	Feature #3	Feature #4	Feature #5	Feature #6	Feature #7	Feature #8	W1	W2	W3	W4	W5	W6	W7	W8
Prompt #1	Direct Reflections	Total Reflection	SDR Brightness	Input Lag	HDR Brightness	VRR	Viewing Angle	Response Time	0.18	0.16	0.15	0.14	0.1	0.1	0.09	0.08
Prompt #2	SDR Brightness	Total Reflection	Contrast	Direct Reflections	Response Time	Viewing Angle	Total Judder	Upscaling	0.18	0.18	0.13	0.12	0.12	0.12	0.08	0.07
Prompt #3	Direct Reflections	Total Reflection	Contrast	Viewing Angle	HDR Brightness	Input Lag	SDR Brightness	VRR	0.18	0.16	0.14	0.12	0.12	0.12	0.08	0.08
Prompt #4	Contrast	Unifrom Blackness	HDR Brightness	HDR Color Volume	HDR Color ACC (Post)	Viewing Angle	Total Reflection	Upscaling	0.18	0.17	0.15	0.13	0.12	0.1	0.08	0.07
Prompt #5	Upscaling	Resolutions	Contrast	SDR Brightness	HDR Color Volume	Viewing Angle	HDR Brightness	LQ Smoothing	0.15	0.14	0.14	0.12	0.12	0.12	0.11	0.1
Prompt #6	Contrast	Unifrom Blackness	HDR Brightness	HDR Color Volume	HDR Color ACC (Post)	Viewing Angle	Upscaling	Total Judder	0.2	0.18	0.15	0.14	0.12	0.08	0.07	0.06
Prompt #7	Input Lag	Input Lag	VRR	HDR Brightness	Response Time	SDR Brightness	Upscaling	Total Reflection	0.18	0.15	0.14	0.13	0.12	0.1	0.09	0.09
Prompt #8	Viewing Angle	Contrast	Unifrom Blackness	SDR Brightness	Resolutions	Upscaling	Frequency Response	Direct Reflections	0.18	0.18	0.16	0.12	0.1	0.1	0.1	0.06
Prompt #9	SDR Brightness	Total Reflection	Contrast	Direct Reflections	Unifrom Blackness	Viewing Angle	HDR Brightness	Build Quality	0.18	0.17	0.15	0.13	0.12	0.1	0.08	0.07
Prompt #10	Color Saturation	Direct Reflections	Total Reflection	Viewing Angle	SDR Brightness	Unifrom Grayscale	Black Level	Build Quality	0.2	0.15	0.12	0.12	0.12	0.11	0.1	0.08
Prompt #11	Contrast	Unifrom Blackness	HDR Brightness	HDR Color Volume	HDR Color ACC (Post)	Total Judder	Viewing Angle	Upscaling	0.2	0.18	0.16	0.14	0.12	0.08	0.07	0.05
Prompt #12	HDR Brightness	Viewing Angle	Contrast	HDR Color Volume	SDR Brightness	Upscaling	Total Reflection	Resolutions	0.16	0.15	0.15	0.13	0.12	0.12	0.09	0.08
Prompt #13	Upscaling	Contrast	HDR Brightness	HDR Color Volume	Viewing Angle	Resolutions	Total Judder	Direct Reflections	0.14	0.14	0.13	0.13	0.12	0.12	0.11	0.11
Prompt #14	Contrast	HDR Brightness	HDR Color Volume	Viewing Angle	SDR Brightness	Direct Reflections	Upscaling	Unifrom Blackness	0.18	0.16	0.14	0.14	0.12	0.1	0.09	0.07
Prompt #15	SDR Brightness	Viewing Angle	LQ Smoothing	Upscaling	Contrast	Input Lag	Unifrom Blackness	Build Quality	0.18	0.16	0.15	0.14	0.13	0.09	0.09	0.06
Prompt #16	SDR Brightness	Viewing Angle	Contrast	Unifrom Blackness	HDR Brightness	Upscaling	Total Reflection	LQ Smoothing	0.17	0.16	0.15	0.12	0.12	0.1	0.1	0.08
Prompt #17	SDR Brightness	Total Reflection	Stutter	Direct Reflections	Input Lag	Upscaling	Viewing Angle	Resolutions	0.2	0.18	0.18	0.12	0.1	0.1	0.07	0.05
Experts' Labeling and Weighting																
	Feature #1	Feature #2	Feature #3	Feature #4	Feature #5	Feature #6	Feature #7	Feature #8	W1	W2	W3	W4	W5	W6	W7	W8
Prompt #1	Direct Reflections	Input Lag	Total Reflection	SDR Brightness	HDR Bright (Game)	VRR	Black Level	Color Saturation	0.25	0.15	0.15	0.15	0.1	0.1	0.05	0.05
Prompt #2	Direct Reflections	SDR Brightness	Response Time	Viewing Angle	Unifrom Grayscale	Total Reflection	Contrast	Frequency Response	0.2	0.2	0.15	0.15	0.1	0.1	0.05	0.05
Prompt #3	Direct Reflections	Total Reflection	Input Lag	Contrast	HDR Brightness	VRR	LQ Smoothing	SDR Brightness	0.25	0.15	0.15	0.15	0.1	0.05	0.05	0.1
Prompt #4	Contrast	Unifrom Blackness	HDR Color Volume	PQ EOTF ACC	Total Judder	Dimming Precision	HDR Color ACC (Pre)	HDR Native Gradient	0.3	0.15	0.15	0.1	0.1	0.1	0.05	0.05
Prompt #5	LQ Smoothing	Upscaling	SDR Brightness	Viewing Angle	HDR Brightness	Contrast	HDR Color Volume	Frequency Response	0.25	0.2	0.15	0.1	0.1	0.1	0.05	0.05
Prompt #6	Contrast	Unifrom Blackness	Total Judder	SDR Brightness	Viewing Angle	Direct Reflections	HDR Brightness	HDR Color Volume	0.3	0.15	0.1	0.1	0.1	0.1	0.1	0.05
Prompt #7	Viewing Angle	Input Lag	LQ Smoothing	Upscaling	VRR	SDR Brightness	Build Quality	HDR Brightness	0.2	0.2	0.15	0.1	0.1	0.1	0.1	0.05
Prompt #8	Contrast	Unifrom Blackness	PQ EOTF ACC	Dimming Precision	Viewing Angle	HDR Brightness	SDR Brightness	HDR Color ACC (Pre)	0.35	0.2	0.1	0.1	0.1	0.05	0.05	0.05
Prompt #9	Direct Reflections	SDR Brightness	Contrast	Total Reflection	Unifrom Blackness	HDR Brightness	Black Level	Viewing Angle	0.25	0.2	0.15	0.1	0.1	0.1	0.05	0.05
Prompt #10	Direct Reflections	Total Reflection	Viewing Angle	SDR Color ACC (Pre)	SDR Brightness	Color Saturation	Build Quality	Contrast	0.3	0.2	0.15	0.1	0.1	0.05	0.05	0.05
Prompt #11	Contrast	HDR Color Volume	Unifrom Blackness	Viewing Angle	Total Judder	Frequency Response	HDR Brightness	PQ EOTF ACC	0.25	0.15	0.15	0.1	0.1	0.1	0.1	0.05
Prompt #12	LQ Smoothing	Upscaling	Viewing Angle	SDR Brightness	HDR Brightness	Contrast	HDR Color Volume	Frequency Response	0.2	0.2	0.2	0.15	0.1	0.05	0.05	0.05
Prompt #13	Contrast	SDR Brightness	HDR Brightness	Unifrom Blackness	HDR Color Volume	SDR Color ACC (Pre)	Upscaling	LQ Smoothing	0.2	0.15	0.15	0.1	0.1	0.1	0.1	0.1
Prompt #14	LQ Smoothing	Upscaling	SDR Brightness	Viewing Angle	HDR Brightness	Contrast	HDR Color Volume	Frequency Response	0.25	0.2	0.15	0.15	0.1	0.05	0.05	0.05
Prompt #15	Viewing Angle	SDR Brightness	Upscaling	LQ Smoothing	SDR Color ACC (Pre)	Build Quality	Response Time	Frequency Response	0.2	0.15	0.15	0.15	0.1	0.1	0.1	0.05
Prompt #16	Viewing Angle	Contrast	SDR Brightness	Unifrom Blackness	HDR Brightness	LQ Smoothing	Upscaling	HDR Color Volume	0.25	0.2	0.15	0.1	0.1	0.1	0.05	0.05
Prompt #17	SDR Brightness	Direct Reflections	Total Reflection	Upscaling	LQ Smoothing	Viewing Angle	Black Level	Color Saturation	0.25	0.2	0.15	0.1	0.1	0.1	0.05	0.05

Table 1: Feature extraction and weight estimation results for each feature; For the experiment, the proposed framework was instructed to calculate eight features and their corresponding weights.; [Acronyms] SDR: Standard Dynamic Range, HDR: High Dynamic Range, VRR: Variable Refresh Rate, LQ: Low-Quality, Color ACC (Pre/Post) - Color Accuracy (Pre/Post-Calibration), PQ EOTF: Perceptual Quantization Electro-Optical Transfer Function.

	NDCG		Precision		Recall	
k	**D-VectorRAG**	**D-GraphRAG**	**D-VectorRAG**	**D-GraphRAG**	**D-VectorRAG**	**D-GraphRAG**
2	0.85	0.92	0.82	0.88	0.84	0.9
3	0.82	0.89	0.79	0.85	0.81	0.87
4	0.79	0.86	0.76	0.82	0.78	0.84
5	0.76	0.83	0.73	0.79	0.75	0.81
6	0.73	0.8	0.7	0.76	0.72	0.78
7	0.71	0.78	0.68	0.74	0.7	0.76
8	0.68	0.75	0.65	0.71	0.67	0.73

Table 2: Evaluation results of average 'Precision', 'Recall', 'NDCG'

line features diminished, reducing both precision (due to denominator growth outpacing numerator gains) and recall (due to coverage gaps in the baseline's newly included features). Weight distribution disparities further explain the performance gap. D-GraphRAG (JSD = 0.15) aligned closely with the baseline's steep weight decay, preserving high-weight feature priorities. This alignment mitigated performance degradation, particularly at lower k-values. In contrast, D-VectorRAG (JSD = 0.22) exhibited flatter weight distributions, amplifying mismatches with the baseline's sharp decline in feature relevance. For instance, at $k = 8$, D-VectorRAG's precision and recall dropped by 21% and 20%, respectively, versus $k = 2$, while D-GraphRAG's declines were milder (19% for precision, 17% for recall). The inversion of the precision-recall trade-off underscores the experiment's unique design. Traditional systems prioritize either metric as k grows, but here, both declined due to the dual pressure of an expanding relevant set and weight misalignment. This phenomenon highlights the sensitivity of evaluation frameworks to the definition of relevance and suggests that conventional baselines may not fully generalize to automated feature selection methods.

Conclusion

The Dynamic Graph-RAG framework, proposed in this study, addresses the critical challenges of reliability and explainability in LLM-based consumer electronics recommendations. Our core contribution is an architecture that synergistically fuses LLMs, RAG, and KGs. By orchestrating specialized agents and utilizing the FDVI for semantic bridging, D-GraphRAG grounds the LLM's reasoning in a structured, factual knowledge base.The experimental findings demonstrate that the proposed methodology effectively mitigates hallucination phenomena while precisely discerning user intents. This capability enables the generation of transparent and logically consistent recommendations, which serves as a critical foundation for establishing user trust in high-stakes application domains.

References

Anuyah, S.; Bolade, V.; and Agbaakin, O. 2024. Understanding graph databases: a comprehensive tutorial and survey. *arXiv preprint arXiv:2411.09999*.

Balloccu, G.; Boratto, L.; Fenu, G.; Malloci, F. M.; and Marras, M. 2024. Explainable recommender systems with knowledge graphs and language models. In *European Conference on Information Retrieval*, 352–357. Springer.

Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Wang, H.; and Wang, H. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).

He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 639–648.

Holzschuher, F.; and Peinl, R. 2013. Performance of graph query languages: comparison of cypher, gremlin and native access in neo4j. In *Proceedings of the Joint EDBT/ICDT 2013 Workshops*, 195–204.

Huang, Y.; and Huang, J. 2024. A survey on retrieval-augmented text generation for large language models. *arXiv preprint arXiv:2404.10981*.

Kwon, J.; Ahn, S.; and Seo, Y.-D. 2024. RecKG: Knowledge Graph for Recommender Systems. In *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*, 600–607.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474.

Liu, Q.; Zhao, X.; Wang, Y.; Wang, Y.; Zhang, Z.; Sun, Y.; Li, X.; Wang, M.; Jia, P.; Chen, C.; et al. 2024. Large Language Model Enhanced Recommender Systems: A Survey. *arXiv preprint arXiv:2412.13432*.

Lopez-Avila, A.; and Du, J. 2025. A Survey on Large Language Models in Multimodal Recommender Systems. *arXiv preprint arXiv:2505.09777*.

Oh, H.; Kim, K.; Kim, J.; Kim, S.; Lee, J.; Chang, D.-s.; and Seo, J. 2024. Exegpt: Constraint-aware resource scheduling for llm inference. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, 369–384.

Panarin, S. 2025. Comparative Assessment of Large Language Model-Driven Recommendation Systems in Smart Spaces.

Peng, Q.; Liu, H.; Huang, H.; Yang, Q.; and Shao, M. 2025. A survey on llm-powered agents for recommender systems. *arXiv preprint arXiv:2502.10050*.

Qiu, Z.; Luo, L.; Zhao, Z.; Pan, S.; and Liew, A. W.-C. 2025. Graph Retrieval-Augmented LLM for Conversational Recommendation Systems. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 344–355. Springer.

Rajabi, E.; and Etminani, K. 2024. Knowledge-graph-based explainable AI: A systematic review. *Journal of information science*, 50(4): 1019–1029.

Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; and Jiang, P. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 1441–1450.

Tourani, A.; Nazary, F.; and Deldjoo, Y. 2025. RAG-VisualRec: An Open Resource for Vision-and Text-Enhanced Retrieval-Augmented Generation in Recommendation. *arXiv preprint arXiv:2506.20817*.

Wang, M.; Guo, Y.; Zhang, D.; Jin, J.; Li, M.; Schonfeld, D.; and Zhou, S. 2024. Enabling explainable recommendation in e-commerce with llm-powered product knowledge graph. *arXiv preprint arXiv:2412.01837*.

Wang, S.; Fan, W.; Feng, Y.; Lin, S.; Ma, X.; Wang, S.; and Yin, D. 2025. Knowledge graph retrieval-augmented generation for llm-based recommendation. *arXiv preprint arXiv:2501.02226*.

Wang, X.; He, X.; Wang, M.; Feng, F.; and Chua, T.-S. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 165–174.

Wu, L.; Zheng, Z.; Qiu, Z.; Wang, H.; Gu, H.; Shen, T.; Qin, C.; Zhu, C.; Zhu, H.; Liu, Q.; et al. 2024. A survey on large language models for recommendation. *World Wide Web*, 27(5): 60.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Zhang, R.; Bao, H.; Sun, H.; Wang, Y.; and Liu, X. 2016. Recommender systems based on ranking performance optimization. *Frontiers of Computer Science*, 10(2): 270–280.

Zhang, Y.; Qiao, S.; Zhang, J.; Lin, T.-H.; Gao, C.; and Li, Y. 2025. A survey of large language model empowered agents for recommendation and search: Towards next-generation information retrieval. *arXiv preprint arXiv:2503.05659*.