X-KGRank: A Knowledge Graph RAG Framework for Explainable Recommendation via Pattern Mining and Grounded LLM Explanation

Meenakshi Rajpurohit
Dept. of Computer Engineering
San Jose State University
San Jose, CA, USA
meenakshi.rajpurohit@sjsu.edu

Jainish Patel
Dept. of Computer Engineering
San Jose State University
San Jose, CA, USA
jainish.patel@sjsu.edu

Rafael Banchs
Dept. of Computer Engineering
San Jose State University
San Jose, CA, USA
rafael.banchs@sjsu.edu

***Abstract*—Modern recommender systems produce predictions that users cannot interrogate. Collaborative filtering captures behavioural signals but offers no reasoning, while large language models (LLMs) generate fluent explanations that are poorly grounded in a user's history. We present X-KGRank, a knowledge-graph retrieval-augmented framework that pairs a structural collaborative-filtering ranker with LLM-generated explanations conditioned on retrieved graph evidence. From MovieLens-1M (6,040 users, 3,704 items, 988,129 interactions) we construct a heterogeneous knowledge graph of 9,762 nodes and 999,264 edges spanning RATED, HAS_GENRE and CO_RATED relations, persisted in Neo4j. We train a LightGCN ranker with content-aware SBERT initialisation and a rating-weighted BPR objective, and retrieve candidate-specific evidence by enumerating and scoring multiple user-to-item paths, ranked by edge strength discounted by bridge-node degree. On the MovieLens-1M test set under a 99-sample protocol the ranker reaches NDCG@10 $= 0.4128$ (95% CI $[0.4038, 0.4215]$), matching tuned BPR-MF (0.4129) and PureSVD (0.4153) and improving on Popularity (0.2533) and ItemKNN (0.3931). Controlled ablations show that neither SBERT initialisation nor rating weighting contributes measurably, whereas popularity-biased negative sampling costs 0.088 NDCG@10. We further show that popularity-selective routing, adopted from prior work, does not activate on this dataset: at the median popularity split only 0.5% of top-20 candidates fall in the long-tail bucket, and none do under the corrected ranker. Across 120 recommendations generated under both grounded and ungrounded prompts by three instruction-tuned LLMs, knowledge-graph grounding raises the rate at which explanations cite retrieved behavioural evidence from 5% to 46% (Qwen2.5-1.5B) and 38% (Mistral-7B), while a heuristic explanation-quality score in common use saturates and cannot distinguish the two models in either condition.**

***Index Terms*—recommender systems, knowledge graph, retrieval-augmented generation, graph neural networks, large language models, explainability**

I. Introduction

When a user asks a recommender for "a mind-bending thriller with unexpected plot twists," what happens? A content-based system that relies only on genre tags labelled "Thriller" may retrieve both a genuine psychological thriller such as *What Lies Beneath* and an unrelated disaster film such as *Twister*, because both carry the same tag. A collaborative-filtering model ranks items by behavioural similarity but offers no user-specific explanation of why a movie was recommended. An LLM-based recommender can generate fluent explanations, but it hallucinates and is factually unreliable: it may miss release years, invent cast members, or attribute films to the wrong directors. The results have limited interpretability and give no clear signal of why they should be trusted. These failure modes arise directly from our own experimental analysis, illustrated in Fig. 1.

Existing frameworks each address part of this problem but each leaves a critical gap. Graph-based collaborative-filtering methods such as LightGCN [1], KGAT [3], and GraphSAGE [2] provide strong behavioural ranking signals, yet their effectiveness degrades as interaction sparsity increases. LLM-based recommenders such as P5 [4] offer good natural-language reasoning, but they hallucinate and often do not ground recommendations in a user's specific interaction history. Knowledge-graph-based recommenders incorporate structural information into ranking, but they typically use the graph as model input rather than as evidence for explanation generation.

A design principle in prior work is that not every item requires the same level of expensive grounding. For popular items an LLM often already possesses the relevant pretrained knowledge; for long-tail items that knowledge is less reliable, and knowledge-graph evidence becomes more valuable. A popularity-selective routing strategy open prompts for popular candidates, KG-grounded prompts for long-tail candidates focuses retrieval where it is presumed most useful. We adopt this design from K-RagRec [6] and, as one of our findings, measure the rate at which it actually activates.

We present X-KGRank, a knowledge-graph retrieval-augmented framework for explainable recommendation. X-KGRank constructs a heterogeneous knowledge graph in Neo4j, trains LightGCN embeddings initialised from SBERT-derived item representations, mines structural and community-level graph features using node2vec and modularity-based clustering, retrieves and ranks candidate-specific knowledge-graph paths at inference time, and conditions an LLM explanation on the retrieved evidence. Our work makes four

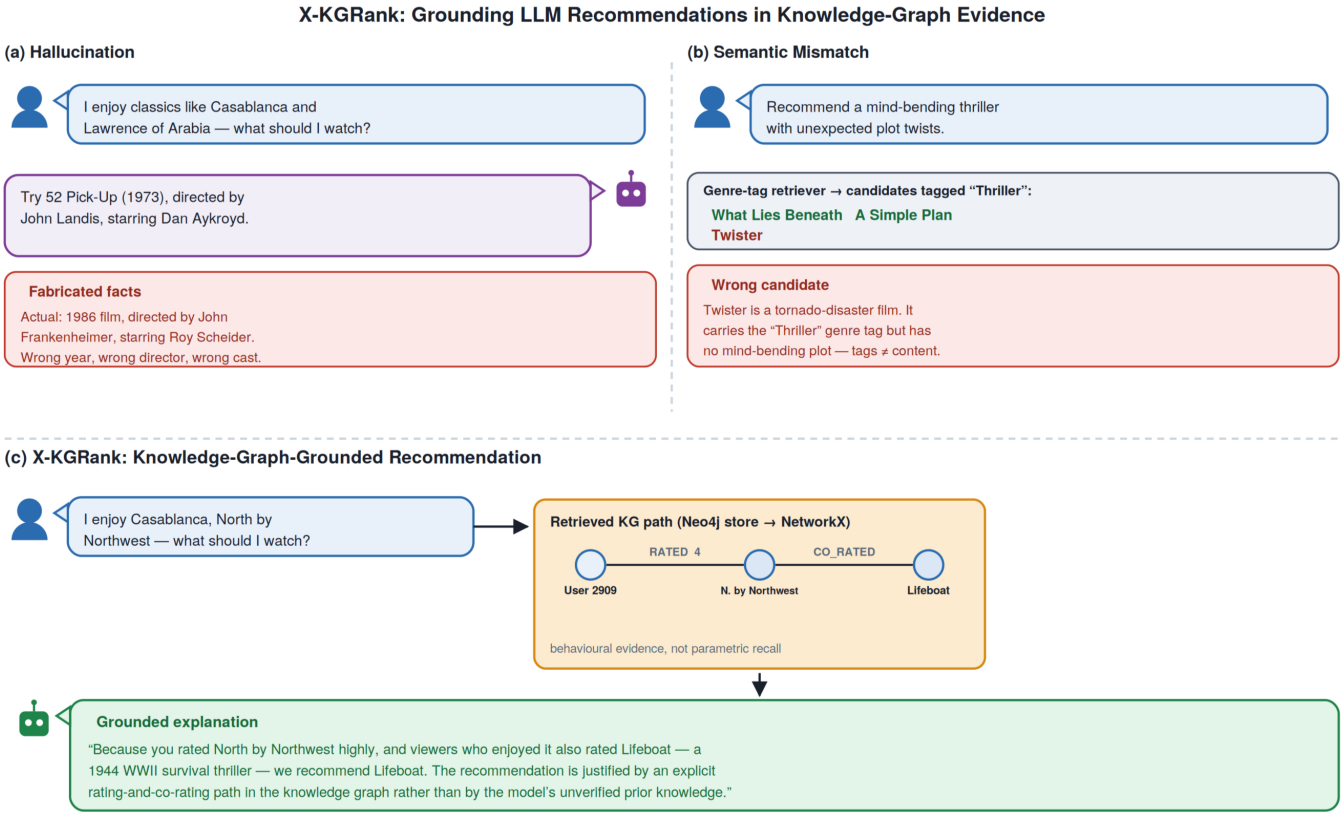


Fig. 1. Two failure modes of LLM-based recommendation and X-KGRank's knowledge-graph-grounded solution. (a) The LLM fabricates a film's director, cast, and year. (b) Genre-tag retrieval returns *Twister*, a disaster film tagged "Thriller," for a mind-bending-thriller query. (c) X-KGRank retrieves an explicit rating-and-co-rating path and conditions its explanation on that behavioural evidence. All examples are drawn from our experiments.

contributions: (i) an end-to-end, openly released framework for knowledge-graph retrieval-augmented explanation over a competitive collaborative-filtering ranker; (ii) a multi-path evidence-aggregation retriever with a degree penalty that removes the hub-path degeneracy of shortest-path retrieval; (iii) a measured analysis of popularity-selective routing showing that, on MovieLens-1M, an accuracy-optimised ranker surfaces essentially no long-tail candidates, so selective retrieval never activates; and (iv) an explanation-quality and factual-reliability comparison across three LLM backbones under both grounded and ungrounded prompts, which identifies a saturation failure in the heuristic metrics commonly used for this purpose.

II. Related Work

The literature relevant to X-KGRank spans three intersecting lines of research: graph neural networks for structural collaborative filtering, LLM-based recommendation, and retrieval-augmented generation over knowledge graphs.

A. Graph Neural Networks for Recommendation

Graph neural networks have become a dominant framework for collaborative filtering, treating user item interactions as a graph and propagating embeddings through multi-hop neighbourhoods. He et al. [1] introduced LightGCN, which simplifies standard graph convolution by removing non-linear activations and feature transformations, retaining only neighbourhood aggregation. We adopt LightGCN as our ranker and add LLM-based explanations that operate on retrieved KG paths. We note that on dense benchmarks evaluated with sampled protocols, carefully tuned matrix factorisation frequently matches graph-based models [10]; our own results in Section VI-A reproduce this pattern.

B. LLMs for Recommendation

Recent work explores LLMs as recommenders. Geng et al. [4] proposed P5, unifying five recommendation tasks sequential recommendation, rating prediction, explanation generation, review-related tasks, and direct recommendation under a single text-to-text framework. While flexible, the pure-LLM approach has two weaknesses: it hallucinates, and it lacks grounding in user-specific behavioural signals. Our work addresses these weaknesses by retaining a structural model for ranking and confining the LLM to an explanatory role over retrieved KG evidence.

C. Retrieval-Augmented Generation over Knowledge Graphs

Retrieval-augmented generation (RAG) has emerged as a remedy for both the sparsity limits of pure structural methods and the hallucination problems of pure-LLM methods. He

et al. [5] proposed G-Retriever, which retrieves task-relevant subgraphs and conditions an LLM on them for textual graph question answering, showing an advantage over dense vector retrieval. Most directly related, Wang et al. [6] introduced K-RagRec, a knowledge-graph retrieval-augmented framework for LLM-based recommendation that applies expensive subgraph retrieval selectively to cold-start items while serving popular items through cheaper structural ranking. X-KGRank builds on the K-RagRec design along three axes. First, we provide an open implementation on MovieLens-1M, including the Neo4j graph database, the LightGCN ranker, the routing analysis, and the MLP projection layer. Second, we replace first-match path retrieval with an evidence-aggregation retriever that enumerates and ranks multiple paths. Third, we evaluate the explanation layer across three LLMs (Flan-T5-large, Qwen2.5-1.5B, Mistral-7B) under both grounded and ungrounded prompts, which isolates the effect of retrieval from the effect of model scale.

III. Preliminaries and Problem Formulation

A. Knowledge Graph and Notation

Let $\mathcal{U} = \{u_1, \ldots, u_n\}$ denote the set of users, $\mathcal{I} = \{i_1, \ldots, i_m\}$ the set of movies, and $\mathcal{C}$ the set of content categories (genres). We model their relationships as a heterogeneous knowledge graph

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \tau), \qquad \mathcal{V} = \mathcal{U} \cup \mathcal{I} \cup \mathcal{C}, \tag{1}$$

where each edge is assigned a relation type by the function $\tau : \mathcal{E} \to \mathcal{R}$ with relation set

$$\mathcal{R} = \{\,\text{RATED},\ \text{HAS_GENRE},\ \text{CO_RATED}\,\}. \tag{2}$$

A RATED edge (u, i) carries an integer rating from 1 to 5; a HAS_GENRE edge (i, c) links an item to a genre; and a CO_RATED edge (i, j) carries a weight w_{ij} equal to the number of users who rated both items.

IV. Methodology

In this section we introduce the key concepts used to build the system and then describe each component of the framework in detail. Fig. 2 gives a high-level overview of the complete pipeline, and Fig. 3 expands each stage with implementation detail.

A. Knowledge Graph Construction

We construct a knowledge graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \tau)$ whose vertex set $\mathcal{V} = \mathcal{U} \cup \mathcal{I} \cup \mathcal{C}$ comprises users, items (movies), and content nodes (genres). Edges are of three types: RATED, HAS_GENRE, and CO_RATED. Fig. 4 shows the resulting schema and the three relation types, and Fig. 5 shows a representative 24-node subgraph.

The RATED relation carries ratings in the range 1–5. The HAS_GENRE edge connects an item to its genre nodes. The CO_RATED edge encodes behavioural co-occurrence and lets the LLM generate the crucial "users who rated A also rated B" explanation; without CO_RATED edges, the only available path between two movies would pass through user nodes, which carry no semantic context for detailed explanations. For example, in MovieLens-1M, User 1007 rated *Raiders of the Lost Ark* (1981) five stars, and the knowledge graph links this behaviour to related items, helping surface user-relevant relationships.

By converting the MovieLens-1M tables into a knowledge graph, Neo4j provides meaning and context that plain lists and tables do not, allowing us to uncover connections useful for recommendation. The full knowledge graph contains 9,762 nodes (6,040 users, 3,704 movies, and 18 genres) and 999,264 edges: 988,129 RATED, 6,190 HAS_GENRE, and 4,945 CO_RATED.

B. Structural Representation Learning

LightGCN propagates information over a graph of users and items connected by rating edges. It removes feature transformation and non-linear activation, keeping only neighbourhood aggregation, which is well suited to user item interactions. Our LightGCN uses 3 propagation layers and an embedding dimension of 128, yielding a model with $1{,}296{,}384$ parameters.

Sentence-BERT (SBERT) turns text into vector embeddings that capture semantic meaning. Each movie is represented using its title and genre, encoded with the `all-MiniLM-L6-v2` SBERT model into a 384-dimensional semantic vector, so movies with similar titles and genres receive nearby representations. Because the recommender uses a 128-dimensional embedding space, a linear projection layer maps each 384-dimensional SBERT vector into the LightGCN dimension. Rather than initialising item embeddings randomly, we therefore start from embeddings that already encode semantic information about each movie. This design is intended to target the cold-start problem; Section VI-A reports an ablation which finds no measurable benefit on this dataset. The item score for a user is the inner product of their learned embeddings,

$$s(u, i) = \mathbf{e}_u^\top \mathbf{e}_i, \tag{3}$$

and the ranker is trained with a rating-weighted BPR objective using Adam (lr $= 10^{-3}$, batch 8,192, $L_2 = 10^{-4}$) for 200 epochs, with a full propagation pass recomputed for every mini-batch and negatives sampled uniformly over the item set. Validation NDCG@10 reaches 0.4292 and is still improving at epoch 200, so the figures we report are a lower bound for this architecture.

Two implementation choices at this stage materially affect results and are ablated in Section VI-A. First, propagation must be recomputed per mini-batch: recomputing once per epoch and reusing the resulting embeddings across optimisation steps applies gradients computed at stale parameter values, and costs 0.03 NDCG@10 in our setting. Second, the negative sampler must be unbiased: drawing half of the negatives from the 200 most popular items trains the model to rank popular items down, which is directly counter-productive under a sampled evaluation protocol whose negatives are drawn uniformly, and costs 0.088 NDCG@10. We report these because both are easy to introduce inadvertently and neither is visible in the training loss.

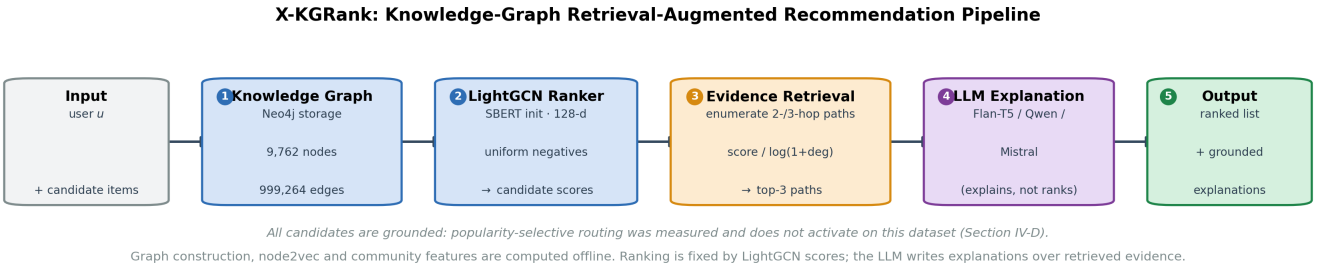


Fig. 2. High-level overview of the X-KGRank pipeline. A user and a candidate item set enter a LightGCN ranker trained over a knowledge graph stored in Neo4j. Retrieved knowledge-graph evidence conditions the LLM explanation, while the final ranking is determined by LightGCN scores.

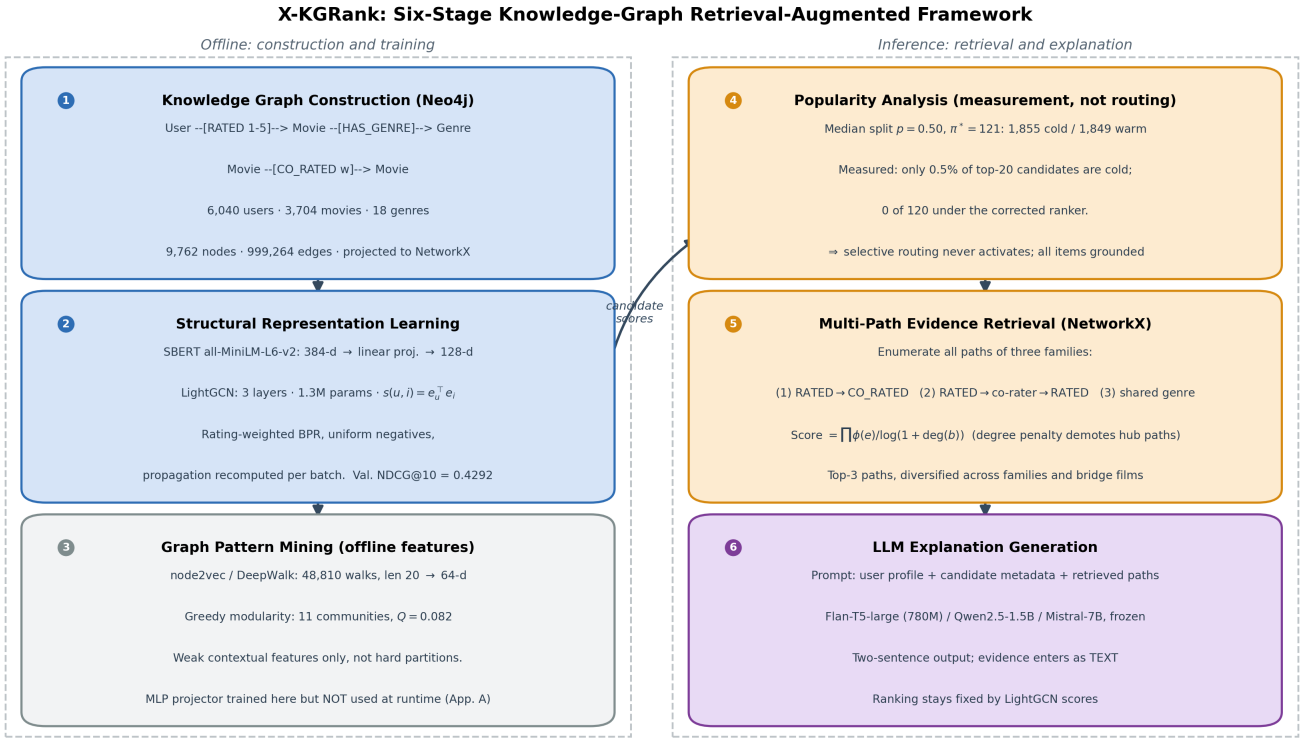


Fig. 3. Detailed X-KGRank architecture. Offline (left): (1) a heterogeneous knowledge graph is stored in Neo4j; (2) a LightGCN ranker is trained with SBERT-initialised embeddings; (3) node2vec and community features are mined. At inference (right): (4) candidate popularity is computed and analysed (Section IV-D); (5) multiple user-to-candidate paths are enumerated and ranked over an in-memory NetworkX projection; (6) the top-ranked paths are serialised into a text prompt for explanation, while ranking stays fixed by LightGCN scores. The MLP projector of Appendix A is trained offline but is not used at runtime.

C. Graph Pattern Mining

LightGCN learns from direct user movie interactions but does not capture long-range graph patterns. We use two graph-mining techniques: node2vec/DeepWalk embeddings and greedy modularity-based community detection.

node2vec performs random walks through the graph, traversing edges in sequence (e.g., user $\rightarrow$ movie $\rightarrow$ user). From 48,810 random walks of length 20, a skip-gram model with negative sampling produces 64-dimensional vectors for all 9,762 nodes. Movies with similar graph structure receive similar embeddings even when no single user rated both, giving the system structural knowledge beyond direct ratings.

We additionally apply greedy modularity maximisation (Clauset–Newman–Moore) and find 11 communities with modularity $Q = 0.082$. The size distribution is bimodal: two large communities—one dominated by movies and one by users—hold 95.1% of all nodes, while the remaining nine range from 446 nodes down to single digits. This reflects the underlying bipartite topology. We do not interpret these communities as definitive categories; they serve only as weak contextual features (Fig. 6).

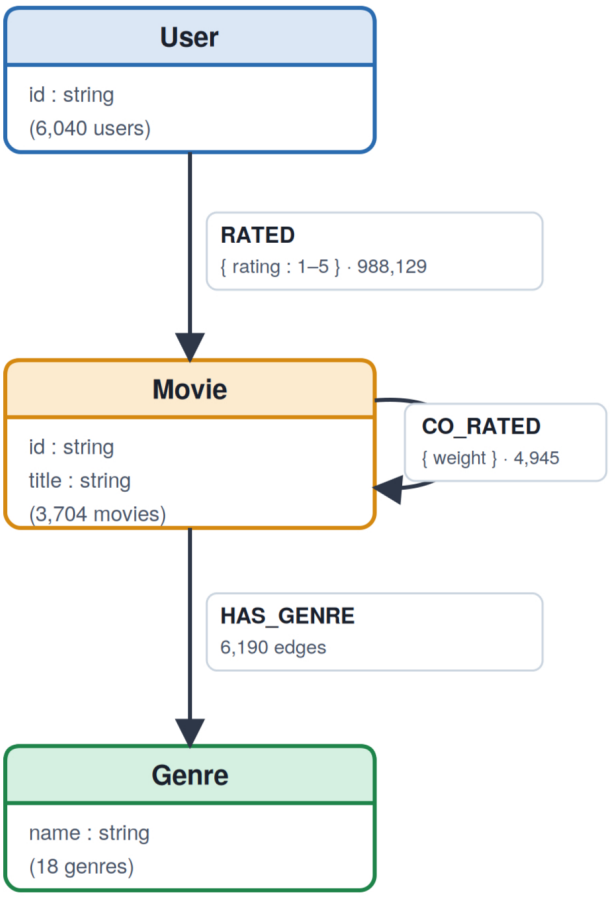


Fig. 4. Schema of the X-KGRank knowledge graph. Three node types (User, Movie, Genre) are connected by three relation edges: RATED (User → Movie), HAS_GENRE (Movie → Genre), and CO_RATED (Movie → Movie).

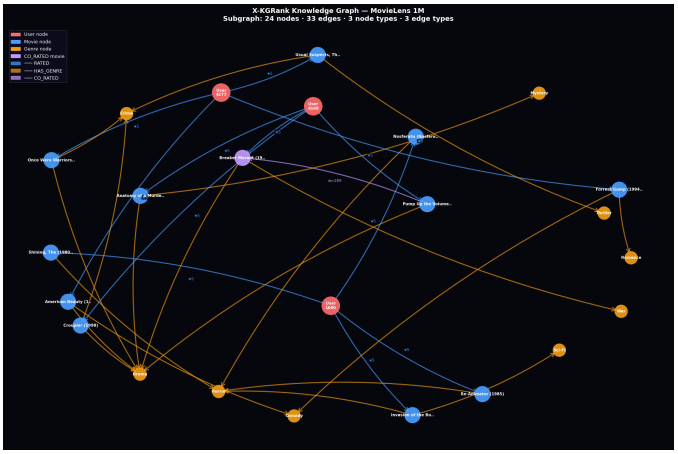


Fig. 5. A 24-node subgraph of the X-KGRank knowledge graph showing User, Movie, and Genre nodes connected by RATED, HAS_GENRE, and CO_RATED edges.

TABLE I
ROUTING-THRESHOLD SWEEP. COLD ITEMS ARE THOSE WITH $\pi(i) \leq$ THE p-TH QUANTILE. "KG RATE" IS THE MEASURED FRACTION OF TOP-20 RANKER CANDIDATES THAT WOULD TRIGGER RETRIEVAL, OVER 500 SAMPLED TEST USERS.

p	π^*	Cold	Cold share		KG rate
			items	inter.	
0.10	7	392	0.106	0.001	0.000
0.25	32	931	0.251	0.012	0.000
0.40	74	1,488	0.402	0.041	0.001
0.50	**121**	**1,855**	**0.501**	**0.076**	**0.005**
0.60	186	2,224	0.600	0.134	0.020
0.75	347	2,779	0.750	0.277	0.097
0.90	722	3,333	0.900	0.555	0.549

D. Popularity-Selective Routing: Premise and Measurement

For very popular items, an LLM has typically already learned the relevant facts from pretraining; for less popular long-tail items, the knowledge graph provides concrete behavioural evidence that the model lacks. Applying KG retrieval only where it is useful would make the system more efficient. We define item popularity $\pi(i)$ as the number of training interactions involving item i and split the catalogue at the p-th quantile of π:

$$\mathcal{I}_{\text{cold}} = \{\, i : \pi(i) \leq \pi^* \,\}, \qquad \mathcal{I}_{\text{warm}} = \{\, i : \pi(i) > \pi^* \,\}. \quad (4)$$

The median split ($p = 0.50$, $\pi^* = 121$ ratings) yields 1,855 cold and 1,849 warm items.

Item counts, however, are the wrong unit of account: what determines cost is how often a *retrieved candidate* is cold. Table I reports, for each threshold, the share of items and of interactions in the cold bucket together with the measured fraction of top-20 ranker candidates that would trigger KG retrieval, over 500 sampled test users.

The measurement contradicts the premise of selective routing on this dataset. At the median split, cold items hold 50.1% of the catalogue but only 7.6% of interactions, and just 0.5% of top-20 candidates; the threshold must be raised to $p = 0.90$ before retrieval fires on even half of candidates. With the corrected ranker the effect is starker still: across 120 cases (40 users × 3 recommendations), *zero* candidates fell below the median. An accuracy-optimised collaborative-filtering ranker concentrates its top-K on head items almost by construction, so a routing rule keyed to item popularity never activates.

We therefore do not use popularity routing in the reported pipeline and ground every candidate. This is also inexpensive: path enumeration over the in-memory NetworkX projection costs milliseconds, against seconds for LLM generation, so the retrieval savings that motivate selective routing are negligible relative to inference cost. We report the analysis because the routing strategy is inherited from K-RagRec [6] and the negative result is informative: selective retrieval is worthwhile only where the ranker is itself tuned for long-tail exposure, or where retrieval is expensive for example remote subgraph queries against a hosted graph database rather than in-memory.

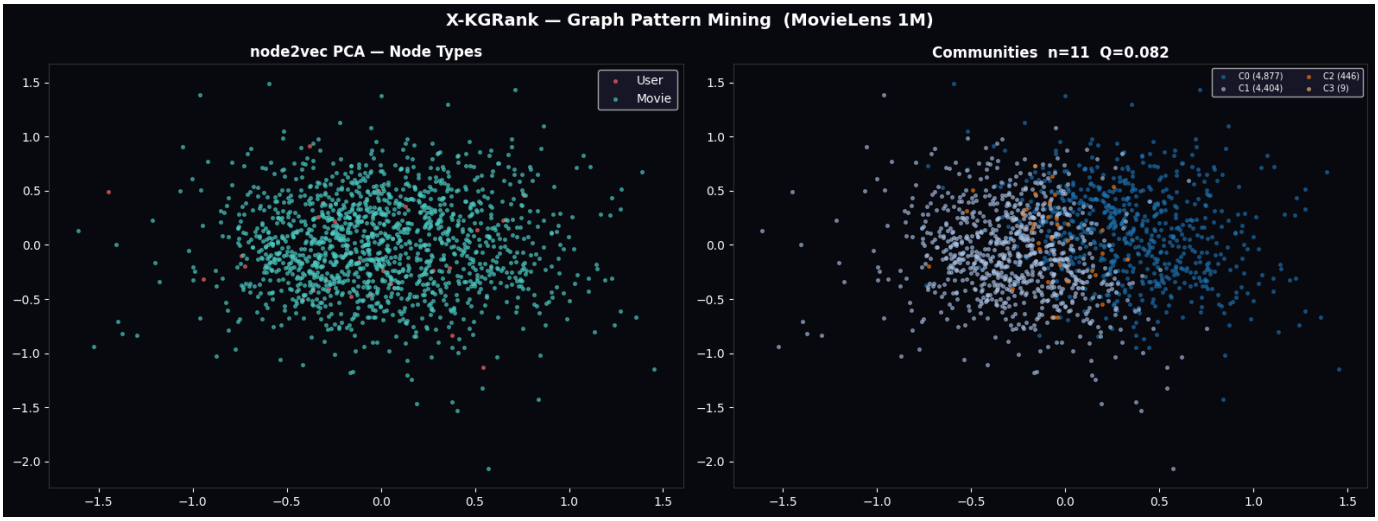


Fig. 6. Graph pattern mining. Left: PCA of node2vec embeddings by node type. Right: 11 communities from greedy modularity maximisation ($Q = 0.082$); the two largest hold 95.1% of nodes.

E. Multi-Path Evidence Retrieval

A good explanation should cite real evidence: that the user rated a particular film highly, that two films are connected by a CO_RATED edge, or that other users who liked the same film also rated the candidate. Rather than accepting the first path a cascade finds, we enumerate all candidate paths and rank them, because the shortest path is frequently not the most informative one.

For a user u and candidate i, we enumerate three families of paths through the user's 15 most highly rated training items m: (1) a two-hop behavioural bridge $u \xrightarrow{\text{RATED}} m \xrightarrow{\text{CO_RATED}} i$; (2) a three-hop collaborative bridge $u \xrightarrow{\text{RATED}} m \xleftarrow{\text{RATED}} v \xrightarrow{\text{RATED}} i$ through a co-rating user v; and (3) a two-hop content bridge through a genre shared by m and i. Each path is scored by the product of its normalised edge strengths ratings scaled to $[0, 1]$, CO_RATED weights as $\log(1+w)$ divided by $\log(1 + \deg(b))$ for the bridge node b:

$$\text{score}(P) = \frac{\prod_{e \in P} \phi(e)}{\log\big(1 + \deg(b_P)\big)}, \tag{5}$$

where $\phi(e)$ is the normalised strength of edge e and b_P the bridge node of path P. The top-k paths ($k = 3$), diversified across the three families and across distinct bridge films, are serialised into the prompt.

The degree penalty addresses a failure mode of shortest-path retrieval documented in Section VII-B: for users with sparse histories, the shortest path repeatedly routes through the same high-degree hub node regardless of the candidate, producing evidence that does not discriminate between candidates. Penalising by bridge degree demotes those hubs in favour of lower-degree, candidate-specific bridges. When no path of any family exists, the system emits an explicit embedding-similarity placeholder so that generation never proceeds ungrounded and the absence of evidence is visible in the output.

The LLM receives a prompt containing the user profile (their five most highly rated training items), the candidate metadata, and the retrieved evidence. The prompt requests a fixed two-sentence output, which keeps inference cost low and forces the model to state only the main reasons. We compare three instruction-tuned LLMs under the same prompt: Flan-T5-large (780M parameters, encoder-decoder), Qwen2.5-1.5B-Instruct (1.5B, decoder), and Mistral-7B-Instruct-v0.2 (7B, decoder). For Flan-T5 we use beam search with 4 beams; for Qwen and Mistral we use greedy decoding. The final ranking is determined by the ranker, so the LLM's role is explanation over the retrieved evidence.

V. Experimental Setup

A. Dataset

We evaluate on MovieLens-1M, which contains roughly one million 1–5 star ratings from $6{,}040$ users on $3{,}704$ movies. The dataset is partitioned into $988{,}129$ training, $6{,}040$ validation, and $6{,}040$ test interactions under leave-one-out splitting, with each user contributing at least 20 ratings (5-core filtering). Ranking metrics are computed over the $6{,}038$ test users whose held-out item is present in the training catalogue. Table II summarises the dataset statistics.

B. Evaluation Protocol

We adopt a 99-sample protocol [7]: for each user in the test set, the held-out positive item is paired with 99 sampled negatives, yielding a 100-item ranking pool. All methods are evaluated on identical pools generated once with a fixed seed, so differences reflect the models rather than the sampling. Confidence intervals are obtained by bootstrap over per-user NDCG@10 with $2{,}000$ resamples, and differences between methods are tested with paired permutation tests over the shared users.

TABLE II
DATASET AND KNOWLEDGE-GRAPH STATISTICS (MOVIELENS-1M).

Statistic	Value
Users	6,040
Movies	3,704
Genres	18
Avg. ratings per user	165.4
Train interactions	988,129
Validation interactions	6,040
Test interactions (evaluated)	6,040 (6,038)
KG nodes	9,762
KG edges	999,264
RATED	988,129
HAS_GENRE	6,190
CO_RATED	4,945
Bipartite graph density	0.088

C. Metrics

We report four standard ranking metrics: (a) NDCG@K; (b) Recall@K; (c) HR@K; and (d) MRR. We report NDCG and Recall at $K \in \{5, 10, 20\}$ and HR at $K = 10$.

For explanation quality we report a heuristic proxy score in $[0, 1]$ combining four equally weighted components ($w = 0.25$ each), computed on the generated text e for a recommendation with title t:

- *Length adequacy*: $\min(|\text{words}(e)|/40, 1)$, where 40 words is the approximate length of a complete two-sentence explanation under our prompt.
- *Reference*: 1.0 if the first eight characters of t appear in e (case-insensitive), else 0.3.
- *Structure*: $\min(n_{\text{sent}}/2, 1)$, where n_{sent} is the number of sentences longer than ten characters after splitting on terminal punctuation.
- *Reasoning*: $\min(n_{\text{kw}}/4, 1)$, where n_{kw} counts distinct reasoning keywords present, matched on word boundaries, from {because, since, as, fan, enjoy, similar, theme, genre, watch, love}.

We additionally report a *citation rate*: the fraction of explanations naming at least one bridge entity an intermediate film or co-rating user that appears only in the retrieved evidence, and not in the user profile or the candidate title. Unlike the quality score, this measures whether the model used the evidence rather than only whether the output is well-formed. No LLM judge is used. The quality score is a proxy for form, not factual accuracy: it cannot detect a fluent explanation containing fabricated cast or dates, and Section VII-C documents exactly such cases scoring at ceiling. We therefore also report a manual factual check.

D. Baselines

We compare against five baselines, all evaluated on identical candidate pools. *Random* assigns uniform random scores. *Popularity* scores each candidate by its training interaction count. *ItemKNN* scores by summed cosine item–item similarity to the user's training items, retaining the 100 nearest neighbours per item. *PureSVD* [8] takes a rank-k truncated SVD of the binary interaction matrix ($k \in \{32, 64, 128, 256\}$; we report $k = 64$, the best on validation). *BPR-MF* [9] is matrix factorisation trained with the pairwise BPR loss at the same embedding dimension, learning rate and regularisation as our ranker, trained to convergence (validation peak at epoch 110). We additionally report three ablations of our own ranker in Table IV. Note that the structural ranker referred to in earlier versions of this work as "LightGCN+SBERT" is not a separate baseline here: because the LLM generates explanations and does not alter the ordering, the ranking metrics of X-KGRank in Table III *are* those of the LightGCN+SBERT ranker evaluated alone.

E. LLM Backbones

The explanation stage is evaluated with three instruction-tuned LLMs: Flan-T5-large, Qwen2.5-1.5B-Instruct, and Mistral-7B-Instruct-v0.2. The MLP projector of Appendix A is trained but is *not* used at inference and does not affect any result reported here.

F. Implementation Details

All stages are implemented in PyTorch and PyTorch Geometric, with HuggingFace Transformers for the LLM backbones, `sentence-transformers` for SBERT encoding, NetworkX for graph construction and traversal, and FAISS for nearest-neighbour retrieval. Ranker training, node2vec/DeepWalk training, and the MLP projector are trained on a single NVIDIA A100 GPU via Google Colab Pro in fp32, and LLM inference is performed in fp16. Code, trained checkpoints, and configuration files are released at https://github.com/MeenakshiRajpurohit/graph-rag-recommend.

VI. RESULTS

A. Ranking Performance

Table III reports ranking performance on the MovieLens-1M test set under the 99-sample protocol, over the 6,038 evaluated test users.

Popularity is a strong baseline on MovieLens-1M and all learned methods clearly exceed it. Among the learned methods, PureSVD (0.4153, CI $[0.4062, 0.4242]$), BPR-MF (0.4129, $[0.4043, 0.4213]$) and our ranker (0.4128, $[0.4038, 0.4215]$) are statistically indistinguishable, while ItemKNN trails slightly (0.3931, $[0.3840, 0.4019]$). This is consistent with a well-documented pattern: on dense MovieLens-scale data with a sampled protocol, carefully tuned matrix factorisation matches graph-based models. We therefore do not claim a ranking advantage for the graph ranker. What the framework contributes is that the same graph which backs the ranker also supplies retrievable, human-readable evidence for explanation which none of the matrix-factorisation baselines can provide at ranking quality that gives up nothing.

Table IV isolates three design choices. SBERT initialisation (0.4128 vs. 0.4107 with random initialisation) and rating-weighted BPR (0.4128 vs. 0.4164 unweighted) have no measurable effect: all confidence intervals overlap, and the unweighted variant is in fact marginally higher. We report

TABLE III

RANKING PERFORMANCE ON MOVIELENS-1M UNDER THE 99-SAMPLE PROTOCOL (6,038 TEST USERS, IDENTICAL CANDIDATE POOLS). BEST PER COLUMN IN **BOLD**; † MARKS METHODS WHOSE 95% BOOTSTRAP CI OVERLAPS THAT OF THE BEST.

Method	NDCG@5	NDCG@10	NDCG@20	Recall@5	Recall@10	Recall@20	HR@10	MRR
Random	0.0318	0.0482	0.0719	0.0565	0.1080	0.2032	0.1080	0.0525
Popularity	0.2052	0.2533	0.2996	0.3089	0.4578	0.6418	0.4578	0.2136
ItemKNN (k=100)	0.3357	0.3931	0.4342	0.4861	0.6630	0.8249	0.6630	0.3269
PureSVD (k=64)	**0.3646**	**0.4153**	0.4471†	**0.5200**	0.6762	0.8014	0.6762	**0.3482**
BPR-MF	0.3564†	0.4129†	**0.4504**	**0.5200**	0.6949†	**0.8425**	0.6949†	0.3411†
X-KGRank (ours)	0.3554†	0.4128†	0.4499†	0.5190†	**0.6958**	0.8413†	**0.6958**	0.3406†

TABLE IV

ABLATIONS OF THE X-KGRANK RANKER. NDCG@10 WITH 95% BOOTSTRAP CIS OVER 6,038 TEST USERS; Δ IS THE CHANGE FROM THE FULL MODEL.

Variant	NDCG@10	95% CI	Δ
Full model	0.4128	[.4038, .4215]	—
w/o SBERT init	0.4107	[.4016, .4195]	−0.002
w/o rating weights	0.4164	[.4075, .4254]	+0.004
w/ biased negatives	0.3250	[.3166, .3336]	−0.088

TABLE V

LLM BACKBONE COMPARISON OVER $N = 120$ SHARED CASES, EACH GENERATED WITH AN OPEN AND A GROUNDED (KG) PROMPT. *Citation* IS THE FRACTION OF EXPLANATIONS NAMING A BRIDGE ENTITY PRESENT ONLY IN THE RETRIEVED EVIDENCE; *Quality* IS THE HEURISTIC SCORE OF SECTION V-C; *Lat.* IS MEAN SECONDS PER EXPLANATION.

Model	Citation		Quality		Lat.
	open	KG	open	KG	
Flan-T5-large	0.042	0.083	0.511 [.495,.527]	0.477 [.457,.499]	2.1
Qwen2.5-1.5B	0.058	**0.458**	0.980 [.972,.988]	0.954 [.944,.964]	4.7
Mistral-7B	0.050	0.375	0.930 [.918,.941]	0.941 [.930,.952]	6.4

this as a negative result. Both choices are motivated by cold-start and preference-strength arguments that plausibly matter on sparser data, but on MovieLens-1M where every user has at least 20 ratings they do not. By contrast, popularity-biased negative sampling is actively harmful, costing 0.088 NDCG@10, because it trains the model against precisely the items that the uniform-negative evaluation protocol rewards.

B. LLM Backbones and the Effect of Grounding

We evaluate the three backbones on $N = 120$ (user, recommendation) cases drawn from 40 users across two random samples, holding the ranker, the retrieved evidence and the prompt template fixed. Each case is generated twice: once with an *open* prompt containing only the user profile and candidate metadata, and once with a *grounded* prompt additionally containing the top-3 retrieved KG paths. Because the two conditions share the same cases, the difference isolates the effect of grounding from the effect of the backbone. The 40 users comprise two independent samples of 20 drawn with different random seeds; per-seed means agree to within 0.021 quality for every model and condition, and all intervals reported below are 95% bootstrap intervals over the pooled 120 cases.

Grounding changes what the models say. Under open prompts all three models cite a bridge entity in roughly 5% of cases, the rate expected by coincidence. Under grounded prompts the rate rises to 45.8% for Qwen2.5-1.5B (+0.400, 95% CI $[+0.317, +0.492]$, $p < 0.0001$) and 37.5% for Mistral-7B (+0.325, $[+0.242, +0.408]$, $p < 0.0001$). Flan-T5-large does not respond to the evidence (8.3% vs. 4.2%, $p = 0.061$). Retrieved graph evidence therefore reaches the output of instruction-tuned models of both sizes, and the behavioural paths that justify a recommendation appear in the text a user reads.

The heuristic quality score cannot measure this. Under the quality score of Section V-C, grounding *decreases* quality for Qwen (0.954 vs. 0.980, $p = 0.0001$) and Flan-T5 (0.477 vs. 0.511, $p = 0.006$), and leaves Mistral unchanged (0.941 vs. 0.930, $p = 0.108$). This is an artifact of the reasoning-keyword component: grounded prompts induce an evidential register ("Based on the evidence provided…") containing fewer of the metric's keywords, so the metric penalises precisely the behaviour the framework is designed to produce. Removing that component eliminates the effect entirely (Qwen $d = 0.000$, $p = 1.0$; Mistral $d = +0.002$, $p = 0.701$; Flan-T5 $d = -0.004$, $p = 0.751$). The remaining components then saturate: Qwen scores 1.000 in both conditions and Mistral 0.992–0.994, so the score has no resolving power between the two large models. We report this as a caution about automatic explanation metrics of this family, which measure surface form and can be maximised by fluent, evidence-free text.

Per-case scores appear in Fig. 7 and the condition summary in Fig. 8.

Model comparison. Both large models decisively outperform Flan-T5-large in both conditions (differences of 0.42–0.48, all $p < 0.0001$). Between them, Qwen2.5-1.5B scores higher than Mistral-7B under open prompts (+0.051, $p < 0.0001$) but the two are statistically indistinguishable under grounded prompts (+0.012, $p = 0.106$), and they are similarly close on citation rate. A 1.5B model therefore matches a 7B model on this task at a quarter of the parameters and roughly 70% of the latency, consistent with the hypothesis that supplying factual content externally reduces reliance on

parametric scale. We qualify this in Section VII-C.

Where grounding falls short. Even in the grounded condition, over half of explanations (54.2% for Qwen, 62.5% for Mistral) cite no retrieved bridge entity, instead producing plausible genre- or theme-level reasoning about the candidate alone. Supplying evidence raises the probability that a model uses it but does not compel use, and nothing in the pipeline verifies that an explanation is faithful to the evidence it was given. Additionally, 18.3% of Qwen's grounded explanations cite anonymous co-rater identifiers ("User 694") verbatim technically grounded, but not meaningful to a reader.

A prompt-formatting note affects the weakest backbone: genre metadata was serialised as a list literal (e.g. `['Action', 'Crime', 'Drama']`) in all prompts. The two decoder models ignored the formatting, but Flan-T5-large echoed the literal in 47.9% of its outputs. Because the condition is identical across models the comparison remains controlled, but Flan-T5's absolute score should be read as a lower bound; its dominant failure mode restating prompt fields rather than reasoning is independent of the formatting.

VII. Qualitative Analysis

To illustrate X-KGRank's behaviour concretely, we walk through cases drawn from a natural-language RAG query demonstration, then examine factual accuracy directly.

A. Success Case: "Moving drama about family and loss"

For this query, SBERT–FAISS retrieved six candidates; the top three were *My Family* (1995, score 0.60), *Two Family House* (2000, 0.56), and *The Funeral* (1996, 0.54). All three are dramas aligned with the query, indicating that the SBERT semantic match operates correctly. For each candidate, the system extracted a KG path connecting the user to the candidate through a shared dramatic film. These paths surface meaningful behaviour: viewers who rated films like *Sixth Sense*, *E.T.*, and *Schindler's List* also engage with highly dramatic family films. The intermediate users are not semantically meaningful in themselves; they are the structure through which the recommendations acquire behavioural grounding. The Flan-T5-large explanations are weak in this case, restating the query rather than reasoning from the retrieved path.

B. Failure Case: "Mind-bending thriller with unexpected plot twist"

For this query, SBERT–FAISS retrieved *What Lies Beneath* (2000, similarity 0.55), *A Simple Plan* (1998, 0.54), and *Twister* (1996, 0.54). The first two are appropriate thrillers. *Twister*, however, is a tornado-disaster film whose genre includes "Thriller" but whose narrative has no relationship to a mind-bending or twist-driven plot. The retrieved paths revealed a second limitation:

```
  #1 What Lies Beneath (2000)
  User 1 --[RATED 4.0]--> Girl, Interrupted (1999)
  Girl, Interrupted (1999) --[RATED 3.0]--> User 90
#3 Twister (1996)
  User 1 --[RATED 4.0]--> Girl, Interrupted (1999)
  Girl, Interrupted (1999) --[RATED 3.0]--> User 90
```

The paths are identical: both route through the same intermediate film and the same intermediate user. Because the demo user has a limited rating history, shortest-path search repeatedly returns the same high-degree bridge node regardless of the candidate. The LLM compounds the error: for *Twister*, Flan-T5-large generates "*Twister* is a mind-bending thriller with unexpected plot twists," projecting the query words onto the candidate rather than reasoning from evidence.

The multi-path retriever of Section IV-E addresses this case directly: because paths are scored with a penalty on bridge-node degree and diversified across distinct bridge films, the high-degree hub responsible for the identical paths is demoted and distinct candidates receive distinct evidence. Paths under the current retriever are also serialised in full, including the final hop into the candidate:

```
  User 5846 --[RATED 5]--> Pulp Fiction (1994)
  --[CO_RATED w=1198]--> Godfather, The (1972)
User 5846 --[RATED 5]--> Wizard of Oz, The (1939)
  --[RATED 5]--> User 12 --[RATED 5]--> Godfather,
The (1972)
User 5846 --[RATED 5]--> Rebecca (1940)
  --[RATED 5]--> User 1256 --[RATED 5]--> Godfather,
The (1972)
```

The remaining limitation is retrieval over genre tags rather than narrative content, which is what admits *Twister* in the first place.

C. Fluency versus Factual Accuracy

The quality score of Section V-C measures surface form. To test whether it tracks factual accuracy we hand-checked 30 grounded explanations the ten highest-scoring from each backbone, since fluent high-scoring output is where fabrication is most consequential—marking each for errors of year, director, cast, or plot. Seven contained a factual error: three from Qwen2.5-1.5B, two from Mistral-7B, and two from Flan-T5-large (Table VI). The per model differences are not meaningful at this sample size and we draw no conclusion from them. What the check does establish is that the metric does not measure factuality at all: every erroneous explanation from the two decoder models received a quality score of 1.000, and across the sample the mean score of erroneous explanations (0.923) was marginally *higher* than that of error-free ones (0.916).

The most instructive failure is Qwen2.5-1.5B on *Under Siege* (1992), which scored 1.000 and was counted as grounded under our citation measure. The explanation describes the film as combining "elements of horror with intense suspense" it is an action film with no horror content and then asserts that "the user has also watched Under Siege (1992), which further establishes their affinity." The latter claim is impossible by construction: candidates are drawn only from items absent from the user's training and validation history. A KG-grounded system is precisely the setting in which such a claim should not

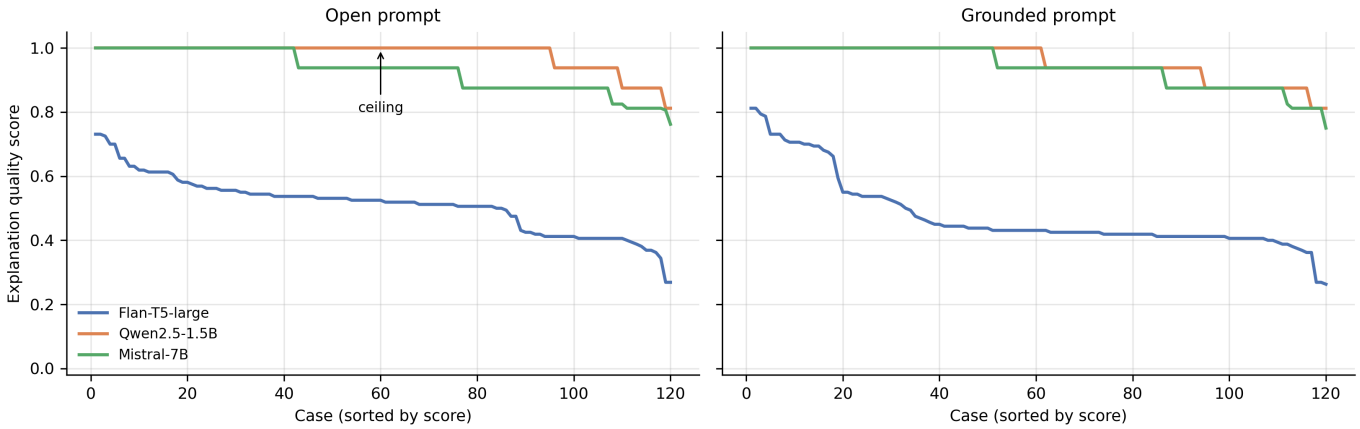


Fig. 7. Per-case explanation quality across the 120 (user, movie) pairs, sorted by score, under open (left) and grounded (right) prompts. The flat upper region for both decoder models is the ceiling effect discussed in Section V-C.

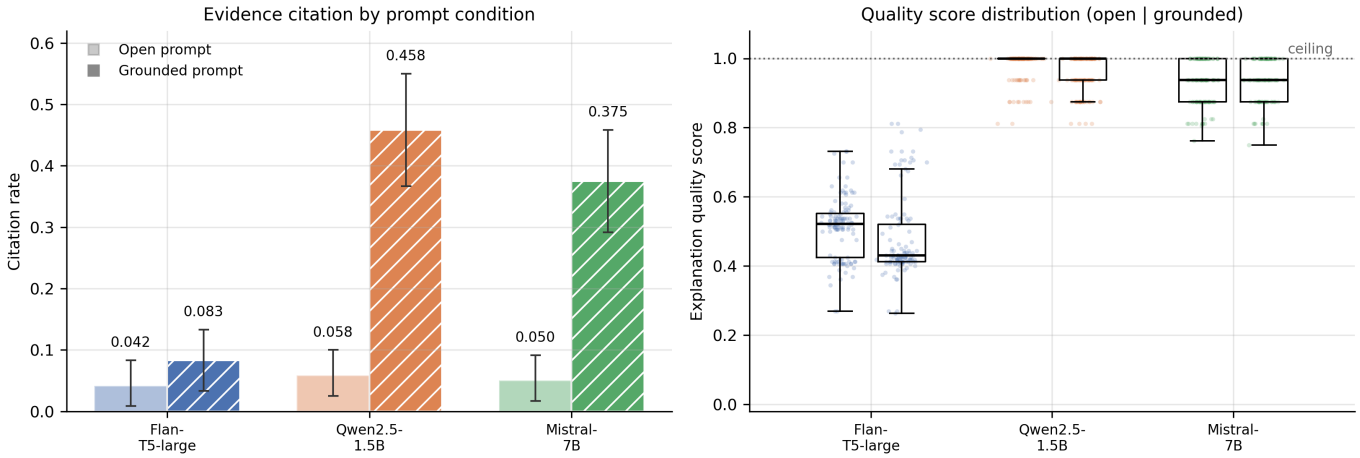


Fig. 8. Left: citation rate by model and prompt condition, with 95% bootstrap confidence intervals. Right: distribution of heuristic quality scores, showing the ceiling effect for the two decoder models.

TABLE VI
MANUAL FACTUAL CHECK OF 30 GROUNDED EXPLANATIONS (THE TEN HIGHEST-SCORING PER BACKBONE). ACROSS ALL 30, ERRONEOUS EXPLANATIONS AVERAGED 0.923 QUALITY VERSUS 0.916 FOR ERROR-FREE ONES.

Model	Errors/10	Categories	Mean qual.
Flan-T5-large	2	year (2)	0.750
Qwen2.5-1.5B	3	plot (2), year (1)	1.000
Mistral-7B	2	plot (2)	1.000

arise, since the user's interaction history is supplied explicitly in the prompt. The model nonetheless fabricated an interaction, and both of our automatic measures scored the output at ceiling.

Mistral-7B's errors are of a different character: it describes the recommended films accurately but mislabels films in the user's profile, calling *The Usual Suspects* a "sci-fi title" in one explanation and a "fantasy film" in another. These are errors about the retrieved context rather than about the recommendation, and are correspondingly less likely to mislead a reader about what they are being recommended.

Flan-T5-large recorded two errors, both corrupted years in text that otherwise restates the prompt verbatim ("*Austin Powers: The Spy Who Shagged Me* (19999)"). Its low error count is not evidence of reliability: the model makes almost no checkable assertions about the recommended film, so there is little for a factual check to find. This illustrates a general hazard in evaluating explanation factuality a model that says nothing substantive cannot be caught being wrong.

Annotation was performed by the authors in a single pass without a second independent annotator; with 30 items and seven errors the counts are indicative rather than precise, and a properly powered study with multiple annotators and inter-annotator agreement remains necessary.

VIII. LIMITATIONS

We analyse the limitations of X-KGRank to contextualise its contributions and guide future work. *(a) Single domain.* All results are obtained on MovieLens-1M; the framework is

unvalidated on e-commerce, news, or sparser domains, and our negative results on SBERT initialisation, rating weighting and popularity routing are specific to a dense 5-core dataset. *(b) No ranking advantage.* Our graph ranker matches but does not beat tuned matrix factorisation; the framework's contribution is the explanation layer. *(c) Genre-tag retrieval.* As the failure case in Section VII-B shows, SBERT–FAISS retrieves over titles and genre tags rather than plot content; richer content representations would mitigate this. *(d) Heuristic explanation metric.* The quality score is uncorrelated with factual accuracy in our manual check, saturates for capable models, and one of its components actively penalises evidential phrasing; the citation rate detects use of evidence but not faithfulness to it. *(e) Evidence is used inconsistently.* Over half of grounded explanations cite no retrieved bridge entity, and the pipeline has no mechanism to require or verify faithful use of the evidence it supplies. *(f) Convergence.* The ranker was still improving at 200 epochs, so the reported ranking figures are a lower bound. *(g) Low community modularity.* The node2vec communities have $Q = 0.082$, reflecting the bipartite topology; we use them only as weak contextual features.

IX. Conclusion

We presented X-KGRank, a knowledge-graph retrieval-augmented framework for explainable recommendation combining a structural collaborative-filtering ranker, multi-path evidence retrieval, and LLM explanation generation grounded in that evidence. On MovieLens-1M the ranker reaches NDCG@10 $= 0.4128$, statistically tied with tuned PureSVD and BPR-MF, so the framework supplies retrievable explanation evidence at no cost in ranking quality. Controlled ablations show that two design choices we had expected to matter content-aware initialisation and rating-weighted BPR do not on a dataset this dense, while unbiased negative sampling matters a great deal. We further find that popularity-selective routing does not activate on MovieLens-1M: an accuracy-optimised ranker surfaces essentially no long-tail candidates, so selective retrieval is worthwhile only where ranking is tuned for long-tail exposure or retrieval is genuinely expensive. Across 120 recommendations generated under both grounded and ungrounded prompts, retrieved evidence raised the rate at which explanations cite behavioural paths roughly eightfold for both Qwen2.5-1.5B and Mistral-7B, while a widely used family of heuristic explanation metrics saturated and could not detect the change. A manual check of 30 grounded explanations found seven factual errors whose mean quality score was marginally higher than that of the error-free ones, including a case in which a grounded explanation fabricated an interaction that the pipeline excludes by construction. Future directions include human evaluation of explanation faithfulness at scale, validation across additional domains, content-attribute knowledge graphs built from plot summaries and reviews, and mechanisms that require rather than merely permit the use of retrieved evidence.

Appendix A
Toward Soft-Prompt Grounding

The knowledge graph provides each movie with a 64-dimensional node2vec vector, whereas the Flan-T5-base encoder operates in a 768-dimensional space. To bridge this gap we train a projector $f_\theta : \mathbb{R}^{64} \to \mathbb{R}^{768}$, a multi-layer perceptron

$$f_\theta(\mathbf{z}) = W_3\,\mathrm{LN}\big(\mathrm{GELU}(W_2\,\mathrm{LN}(\mathrm{GELU}(W_1\mathbf{z})))\big), \quad (6)$$

whose output is normalised to unit length (281,088 trainable parameters). Targets are mean-pooled Flan-T5-base encodings of each movie's title and genre. We use an InfoNCE objective rather than a reconstruction loss, because reconstruction drives the projected vectors toward the average text vector, whereas InfoNCE keeps each movie individually identifiable:

$$\mathcal{L}_{\mathrm{InfoNCE}} = -\log \frac{\exp\big(\mathrm{sim}(f_\theta(\mathbf{z}_i), \mathbf{t}_i)/\tau\big)}{\sum_j \exp\big(\mathrm{sim}(f_\theta(\mathbf{z}_i), \mathbf{t}_j)/\tau\big)}, \quad \tau = 0.07. \quad (7)$$

Training loss falls from 4.01 at epoch 10 to 3.11 at epoch 40; a uniform random projector over a batch of 256 would yield an expected loss of $\log 256 = 5.55$, so the achieved value represents a 44% reduction, indicating a non-trivial alignment.

The projector is not part of the evaluated pipeline. In all experiments reported in Sections VI-A–VII-C the retrieved KG evidence is supplied to the LLM as text, and no projected embedding is used; the projector therefore has no effect on any reported number. We include it as a trained, released component for future work on soft-prompt grounding, where the projected vector would be prepended to the LLM input in place of the serialised path.

Appendix B
Project Artifacts and Implementation Resources

All source code, notebooks, datasets, and visualisations are publicly accessible. These resources cover the full pipeline from knowledge-graph construction and structural representation learning to evidence retrieval and LLM-based explanation together with the per-case generation logs underlying Tables V and VI:

https://github.com/MeenakshiRajpurohit/graph-rag-recommend

References

[1] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, 2020, pp. 639–648.

[2] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 1024–1034.

[3] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT: Knowledge Graph Attention Network for Recommendation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining*, 2019, pp. 950–958.

[4] S. Geng, S. Liu, Z. Fu, Y. Ge, and Y. Zhang, "Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5)," in *Proc. 16th ACM Conf. Recommender Systems (RecSys)*, 2022, pp. 299–315.

[5] X. He, Y. Tian, Y. Sun, N. V. Chawla, T. Laurent, Y. LeCun, X. Bresson, and B. Hooi, "G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[6] S. Wang et al., "K-RagRec: Knowledge Graph Retrieval-Augmented Generation for LLM-based Recommendation," *arXiv preprint* arXiv:2501.02226, 2025.

[7] W. Krichene and S. Rendle, "On Sampled Metrics for Item Recommendation," in *Proc. 26th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining*, 2020, pp. 1748–1757.

[8] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of Recommender Algorithms on Top-N Recommendation Tasks," in *Proc. 4th ACM Conf. Recommender Systems (RecSys)*, 2010, pp. 39–46.

[9] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian Personalized Ranking from Implicit Feedback," in *Proc. 25th Conf. Uncertainty in Artificial Intelligence (UAI)*, 2009, pp. 452–461.

[10] M. F. Dacrema, P. Cremonesi, and D. Jannach, "Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches," in *Proc. 13th ACM Conf. Recommender Systems (RecSys)*, 2019, pp. 101–109.