# Adaptive Candidate Retrieval with Dynamic Knowledge Graph Construction for Cold-Start Recommendation

Anonymous ACL submission

## Abstract

The cold-start problem remains a critical challenge in real-world recommender systems, as new items with limited interaction data or insufficient information are frequently introduced. Despite recent advances leveraging external knowledge such as knowledge graphs (KGs) and large language models (LLMs), recommender systems still face challenges in practical environments: (1) static KGs are costly to construct and maintain and quickly become outdated as catalogs evolve; and (2) LLM-based methods are constrained by limited context windows, forcing reliance on pre-filtered candidate lists. To address these limitations, we propose ColdRAG, a retrieval-augmented framework that dynamically constructs a knowledge graph from raw metadata, extracts entities and relations to construct an updatable structure, and introduces LLM-guided multi-hop reasoning at inference time to retrieve and rank candidates without relying on pre-filtered lists. Experiments across multiple benchmarks show that ColdRAG consistently outperforms seven strong baselines. Our implementation is available at https://anonymous.4open.science/r/ColdRAG.

## 1 Introduction

In real-world recommender systems, newly introduced items often arrive with few or no interaction records and incomplete metadata. This lack of information prevents models from accurately estimating user preferences, resulting in poor recommendation quality, reduced user satisfaction, and ultimately revenue loss (Huang et al., 2023; Zhang et al., 2025). To tackle this challenge, recent works have explored two main directions: (i) KG–based methods that construct structured representations of the item catalog (Wang et al., 2019b; Guo et al., 2020), and (ii) LLM-based methods that leverage LLMs as recommenders, typically prompting on user histories with a small set of candidate items (Sanner et al., 2023; Hou et al., 2024).

However, both directions face critical limitations in practical deployment (Lin et al., 2025). First, knowledge graphs used in prior work are typically constructed offline and treated as static structures(Wang et al., 2019a; Jiang et al., 2024). Building such graphs requires substantial manual engineering to extract and curate relations among items and attributes, making them costly to construct and maintain. Moreover, real-world item catalogs evolve continuously as new items, attributes, and relations appear. Maintaining these static graphs requires periodic offline updates and recalibration, which limits real-time adaptability and prevents end-to-end integration with dynamic recommendation pipelines (Guo et al., 2020).

Second, LLM-based approaches are usually formulated as re-rankers over a pre-filtered candidate set rather than as end-to-end retrieval systems (Hou et al., 2024; Wu et al., 2024). Because LLMs operate under bounded context windows and token budgets, only a curated subset of items and a truncated user history can be included in the prompt, necessitating a separate task-specific retrieval pipeline. While recent work introduces retrieval-aware prompting, its retrieval remains shallow, limited to keyword matching or single-hop similarity search, which still limits performance and reduces adaptability in dynamic cold-start settings (Liang et al., 2025; Kieu et al., 2025).

To address the above limitations, we introduce ColdRAG, a retrieval-augmented framework built around two key modules. The first module, Dynamic Knowledge Graph Construction, automatically builds and incrementally updates a domain graph from catalog fields (e.g., titles, descriptions, attributes, reviews), allowing the structure to evolve naturally as the catalog changes. The second module, Adaptive Candidate Retrieval over Knowledge Graph, removes the need for pre-filtered candidate lists by treating candidate generation as LLM-

guided, goal-directed traversal over the graph, assembling a compact, high-utility candidate set together with evidence paths that justify each recommendation. Empirically, ColdRAG consistently surpasses strong training-based and training-free baselines across diverse product domains with large performance gains.

Our contributions are threefold:

- To the best of our knowledge, we are the first to introduce a dynamic KG evolving mechanism for recommender systems that adapts to an ever-changing item catalog, reducing static KG maintenance and engineering overhead.
- We eliminate the unrealistic assumption that a curated candidate list is already in the LLM's context window by integrating candidate retrieval with LLM-guided multi-hop reasoning.
- We demonstrate strong cold-start performance on multiple benchmarks and provide extensive analyses on component effectiveness, stability, and robustness.

## 2 Related Works

### 2.1 Cold-Start Recommendation

The item cold-start problem arises when newly introduced items lack sufficient interaction history, making it difficult for traditional recommender systems to estimate user preferences (Zhang et al., 2025). Early works focused on content-based or hybrid methods that leverage auxiliary information such as item descriptions or attributes to compensate for missing collaborative signals (Javed et al., 2021; Widayanti et al., 2023). More recent training-based approaches explicitly target cold-start-scenarios by designing specialized objectives or robustness mechanisms. For example, CLCRec (Wei et al., 2021) strengthens cold-start representations through contrastive alignment between collaborative and content-derived embeddings, while TDRO (Lin et al., 2024) improves generalization and robustness by accounting for temporal distribution shifts. Although effective, these methods depend on task-specific training pipelines and require retraining or adaptation as new items continuously appear, which limits their flexibility in dynamic real-world settings.

### 2.2 LLM-based Recommendation

Building on this perspective, a line of work has framed cold-start recommendation as a zero-shot learning problem, requiring generalization to cold items that lack historical interactions (Li et al., 2019; Alshehri and Zhang, 2022). This framing naturally motivates the use of LLMs, as their pretrained semantic knowledge and instruction-following ability allow them to operate in training-free, zero-shot settings using only textual descriptions of items and user histories (Sanner et al., 2023). Following this direction, several zero-shot LLM recommenders have been proposed. LLM-Rank formulates recommendation as an instruction-following task, prompting an LLM to rank candidate items based on a user's interaction history and item descriptions without any additional training (Hou et al., 2024). TaxRec extends this approach by augmenting prompts with taxonomy tokens, enabling the model to reason over hierarchical item spaces (Liang et al., 2025). KALM4Rec further incorporates a lightweight keyword-driven retrieval module that supplies salient terms before prompt-based LLM re-ranking, while still avoiding finetuning (Kieu et al., 2025). Despite their improvements, these methods primarily rely on prompt design and typically operate over pre-filtered candidate lists or shallow retrieval signals, as they are constrained by static knowledge and the limited context window budget of LLMs.

## 3 Proposed Method

We present **ColdRAG**, a retrieval-augmented generation framework for cold-start recommendation. ColdRAG equips an LLM with a dynamically constructed KG that enables semantic reasoning and adaptively builds candidate items for context-aware and controllable recommendation. The overall framework is illustrated in Figure 1. For the problem setting, we follow the sequential recommendation task, where each user $u$ has an interaction history $H_u = [i_1, i_2, \ldots, i_{n-1}]$ and the goal is to recommend the next item $i_n$.

### 3.1 Dynamic Knowledge Graph Construction

#### 3.1.1 Item Profile Generation

Item metadata is a key source for constructing the knowledge graph, but it is often sparse, noisy, or inconsistently structured, making it difficult to extract meaningful semantics directly. To address this, ColdRAG leverages a LLM to denoise and enrich this information by using the model's pretrained knowledge to fill informational gaps, producing concise, knowledge-grounded item pro-

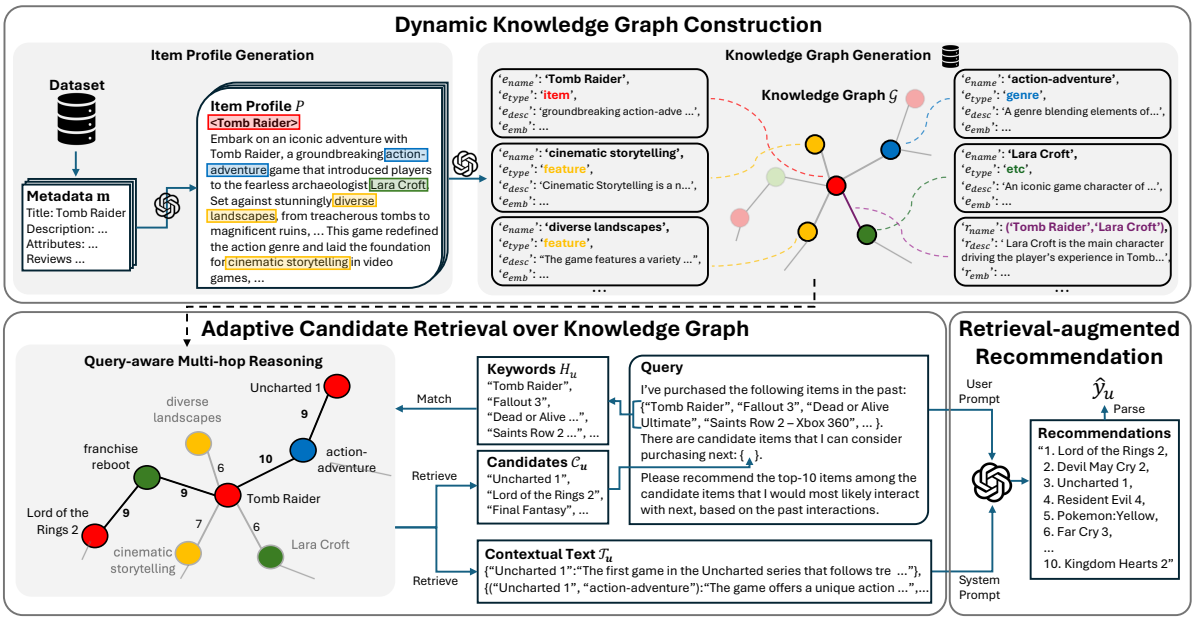


Figure 1: Overview of the proposed **ColdRAG** framework. Given item metadata, an LLM generates item profiles, from which structured entities and relations form a knowledge graph dynamically. During inference, ColdRAG performs query-aware multi-hop reasoning over KG to adaptively retrieve candidate items and context, then composes prompts to generate recommendations.

files that capture each item's essential semantics. For each item $i$, we define its metadata as $\mathbf{m}_i = (\text{title}, \text{description}, \text{attributes}, \text{review})$ and obtain the profile via the inline mapping $P_i = \mathit{LLM}(\mathit{prompt}(\mathbf{m}_i))$. This process curates raw, unstructured metadata into fluent summaries that highlight key concepts such as *genre*, *features*, or *notable entities* (e.g., "action-adventure," "Lara Croft"), forming a clean and standardized foundation for knowledge graph construction and downstream reasoning. The prompt used for this step is provided in Appendix A.1.

### 3.1.2 Knowledge Graph Generation

With enriched item profiles in place, ColdRAG organizes this information into a structured semantic graph by prompting the LLM to extract entities and relations, which creates a foundation for reasoning and retrieval. Given an item profile $P_i$, the LLM extracts entities, relations, and textual statements describing them, constructing a knowledge graph $\mathcal{G} = (\mathcal{E}, \mathcal{R})$, where $\mathcal{E}$ denotes entities and $\mathcal{R}$ the relations linking them.

To be specific, for each item, the LLM extracts entities and relations, each paired with a natural-language description and an embedding for semantic retrieval. An entity $e \in \mathcal{E}$ is represented as $(e_{\text{name}}, e_{\text{type}}, e_{\text{desc}}, e_{\text{emb}})$, where $e_{\text{name}}$ is the entity title, $e_{\text{type}}$ identifies its category (e.g., item, genre, feature), $e_{\text{desc}}$ provides its textual explanation, and $e_{\text{emb}}$ is its vector embedding. A relation $r \in \mathcal{R}$ is represented as $(r_{\text{name}}, r_{\text{desc}}, r_{\text{emb}})$, where $r_{\text{name}} = (e_{\text{src}}, e_{\text{tgt}})$ denotes the source-target entity pair, $r_{\text{desc}}$ describes their connection, and $r_{\text{emb}}$ is its embedding used for scoring. For instance, as shown in Figure 1, the entity Tomb Raider is of type 'item' and the description "groundbreaking action-adventure game ...". A corresponding relation connects Tomb Raider to Lara Croft with the description "Lara Croft is the main character driving the player's experience in Tomb Raider.". This transformation organizes free-form text into an entity-centric graph structure, enabling fine-grained multi-hop reasoning over attribute-level connections. The prompt used is shown in Appendix A.2.

The constructed graph is stored in a hybrid knowledge base: the graph topology (nodes and edges) and their textual descriptions are stored as structured files, while embeddings are indexed in a vector database (e.g., *FAISS*[1]) for efficient similarity search. Each textual description is encoded using the same pretrained embedding model, ensuring consistent semantic representations across all entities and relations; additional details are

[1] https://github.com/facebookresearch/faiss

provided in Appendix B. This combination of structural traversal and semantic retrieval provides ColdRAG with evidence-grounded access to item knowledge during recommendation.

### 3.2 Adaptive Candidate Retrieval over Knowledge Graph

Once the knowledge graph $\mathcal{G}$ has been constructed, ColdRAG uses it to adaptively identify candidate items aligned with a user's current interests. Given a user query that includes both task instructions and the interaction history $H_u$, the system begins by locating the parts of the graph most relevant to the user. The titles of items in $H_u$ are used as keyword anchors and embedded with the same pre-trained embedding model from graph construction. These embeddings are then matched against stored entity embeddings using cosine similarity to locate the most semantically similar nodes. The matched entities initialize the frontier $\mathcal{F}_0$, representing the user's current semantic context in the graph. ColdRAG then performs iterative query-aware multi-hop reasoning guided by the LLM. At each step $t$, all outgoing edges from the current frontier $\mathcal{F}_t$ are scored by the LLM according to their relevance to the user history, where $s_r = \textit{LLM}(r_{\text{desc}}, H_u)$ and $s_r \in [0, 10]$ measures the semantic alignment between the relation description $r_{\text{desc}}$ and the user's interests. Edges with $s_r \geq \lambda$ are retained, and their target nodes form the next frontier:

$$\mathcal{F}_{t+1} = \{e' \mid (e, e', r_{\text{desc}}) \in \mathcal{R},\ s_r \geq \lambda\}.$$

When a target node corresponds to an item, it is added to a temporary candidate pool $\widetilde{\mathcal{C}}_u$ along with its associated descriptions $\widetilde{\mathcal{T}}_u$. Traversal continues until $|\widetilde{\mathcal{C}}_u|$ reaches the predefined maximum pool size $\theta_{\text{pool}}$. Finally, the LLM aggregates edge scores to rank the retrieved items and selects the top $\theta_{\text{top}}$ as the final candidate set $\mathcal{C}_u$, with their textual evidence forming the final contextual input $\mathcal{T}_u$. The example prompt is shown in Appendix A.3.

### 3.3 Retrieval-augmented Recommendation

In the final stage, ColdRAG generates recommendations using the candidate item set $\mathcal{C}_u$ and contextual text block $\mathcal{T}_u$. The contextual text, composed of natural-language descriptions of relevant entities and relations from the knowledge graph, serves as the *system prompt* that provides semantic grounding. The candidate set is integrated into the user query to form the *user prompt* $Q_u$, which expresses user preferences and specifies the desired top-$k$ recommendations within the retrieved candidates.

The LLM then generates ranked outputs conditioned on both prompts:

$$\hat{\mathcal{Y}}_u = \textit{ParseTopK}(\textit{LLM}(\mathcal{T}_u, Q_u, k)).$$

Here, $\textit{LLM}(\mathcal{T}_u, Q_u, k)$ denotes generation with top-$k$ instruction (e.g., "Recommend the top-$k$ items among the given candidate list, based on the user's history and retrieved context"), and *ParseTopK* extracts top-k ranked item titles from the output. ColdRAG's ability to accommodate new items is further discussed in Section 5.1.

Table 1: Summary of dataset and constructed knowledge graph statistics.

| Dataset | #Interactions | #Items | #Users | #Nodes | #Edges |
|---|---|---|---|---|---|
| Games | 45,106 | 2,027 | 2,096 | 15,048 | 29,023 |
| Toys | 332,055 | 12,342 | 20,390 | 58,096 | 132,229 |
| Office | 233,738 | 6,107 | 15,302 | 42,769 | 75,053 |

## 4 Experiments

We conduct comprehensive experiments to evaluate the effectiveness of ColdRAG in item cold-start recommendation. Our analysis is organized around the following research questions:

- **RQ1:** Does ColdRAG effectively address the item cold-start recommendation problem?
- **RQ2:** How effective is the Dynamic Knowledge Graph Construction?
- **RQ3:** How does Adaptive Candidate Retrieval enhance recommendation performance?
- **RQ4:** Does ColdRAG exhibit stable and consistent generation across runs?
- **RQ5:** Does ColdRAG reduce hallucination and avoid out-of-domain recommendations?

### 4.1 Experimental Setup

#### 4.1.1 Datasets

We evaluate ColdRAG on three domains from the Amazon Review dataset (Ni et al., 2019): *Games*, *Toys*, and *Office*, which represent diverse product types and interaction patterns. We apply core filtering with a threshold of 15 for *Games* and 10 for *Toys* and *Office*, retaining users and items that meet the minimum interaction count. To simulate item cold-start scenarios, the least frequent 10% of items in each dataset are designated as *cold items*. Following the sequential recommendation setting, each user's interactions are treated as a sequence,

Table 2: Comparison of Recall@10 and NDCG@10 (%) with **training-free** baselines and ColdRAG (GPT/Qwen) across three datasets. Best results are in bold and second-best baseline is underlined for each LLM. The reported improvement reflects the relative gain of ColdRAG over the strongest competing baseline. All results are averaged over 5 runs and values are shown as percentages.

| LLM | Model | Games | | Toys | | Office | |
|---|---|---|---|---|---|---|---|
| | | **Recall@10** | **NDCG@10** | **Recall@10** | **NDCG@10** | **Recall@10** | **NDCG@10** |
| GPT | LLM | 3.48 | 1.52 | 1.32 | 0.60 | 3.16 | 1.34 |
| | LLMRank (S) (Hou et al., 2024) | 4.62 | 1.94 | 1.36 | 0.58 | 3.80 | 1.63 |
| | LLMRank (R) (Hou et al., 2024) | 8.86 | 4.25 | 1.13 | 0.72 | 4.20 | 2.23 |
| | LLMRank (I) (Hou et al., 2024) | 6.20 | 3.78 | 1.20 | 0.70 | 4.02 | 1.90 |
| | TaxRec (Liang et al., 2025) | 3.75 | 1.78 | 0.60 | 0.34 | 2.20 | 0.96 |
| | KALM4Rec (Kieu et al., 2025) | 8.50 | 4.14 | 4.26 | 2.13 | 3.43 | 1.71 |
| | **ColdRAG** | **12.38** | **4.37** | **5.40** | **2.29** | **8.60** | **3.26** |
| | **Improvement** | 39.73% | 2.82% | 26.76% | 7.51% | 104.76% | 46.18% |
| Qwen | LLM | 9.24 | 3.89 | 0.56 | 0.33 | 3.30 | 1.63 |
| | LLMRank(S) (Hou et al., 2024) | 10.92 | 5.14 | 1.01 | 0.65 | 3.67 | 1.87 |
| | LLMRank(R) (Hou et al., 2024) | 10.98 | 5.39 | 1.20 | 0.71 | 2.27 | 1.42 |
| | LLMRank(I) (Hou et al., 2024) | 9.08 | 4.88 | 1.24 | 0.87 | 4.07 | 2.39 |
| | TaxRec (Liang et al., 2025) | 7.61 | 4.63 | 0.80 | 0.59 | 3.81 | 2.12 |
| | KALM4Rec (Kieu et al., 2025) | 7.27 | 2.62 | 3.29 | 1.90 | 2.07 | 0.58 |
| | **ColdRAG** | **19.57** | **6.50** | **4.10** | **1.98** | **9.40** | **3.92** |
| | **Improvement** | 78.22% | 20.59% | 24.62% | 4.21% | 130.99% | 64.02% |

Table 3: Comparison of Recall@10 and NDCG@10 (%) with **training-based** baselines and ColdRAG (GPT/Qwen) across three datasets. Best results are in bold. All results are averaged over 5 runs and reported as percentages.

| LLM | Model | Games | | Toys | | Office | |
|---|---|---|---|---|---|---|---|
| | | **Recall@10** | **NDCG@10** | **Recall@10** | **NDCG@10** | **Recall@10** | **NDCG@10** |
| – | UniSRec (Hou et al., 2024) | 1.14 | 0.48 | 1.48 | 0.71 | 1.41 | 0.66 |
| – | CLCRec (Wei et al., 2021) | 5.71 | 2.98 | 2.75 | 1.36 | 3.36 | 1.98 |
| – | TDRO (Lin et al., 2024) | 6.61 | 4.21 | 2.64 | 1.31 | 3.79 | 2.14 |
| GPT | **ColdRAG** | 12.38 | 4.37 | **5.40** | **2.29** | 8.60 | 3.26 |
| Qwen | **ColdRAG** | **19.57** | **6.50** | 4.10 | 1.98 | **9.40** | **3.92** |

where the last item is held out for testing under the leave-one-out protocol (Sun et al., 2019; Hou et al., 2022). From each domain, we sample 500 user sequences that end with a cold item; the preceding $n-1$ items serve as input and the final item as the test target. For training-based baselines, these 500 cold-item sequences are used for testing, and all remaining sequences form the training set, ensuring consistent evaluation between training-based and training-free settings. Dataset and knowledge graph statistics are summarized in Table 1.

### 4.1.2 Baselines

We compare ColdRAG with representative baselines spanning both *training-based* and *training-free* paradigms. Among training-based models, **UniSRec** (Hou et al., 2022) fine-tunes a universal sequence encoder with contrastive objectives, **CLCRec** (Wei et al., 2021) trains a contrastive framework to preserve collaborative signals for cold items, and **TDRO** (Lin et al., 2024) applies distributionally robust optimization to handle temporal shifts. For training-free methods, we include a plain **LLM** that ranks randomly sampled candidates without retrieval grounding; **LLMRank** (Hou et al., 2024), which provides sequential (S), recency (R), and in-context (I) prompting variants for zero-shot re-ranking; **TaxRec** (Liang et al., 2025), which injects taxonomy cues to guide LLM reasoning; and **KALM4Rec** (Kieu et al., 2025), which uses keyword-level retrieval to support cold-start recommendation. Among the *training-based* baselines, UniSRec fine-tunes a pretrained model, whereas CLCRec and TDRO are trained from scratch. The remaining methods are *training-free* methods that operate entirely without parameter updates, relying on LLM inference for reasoning and retrieval. Together, they provide a broad comparison across representation learning, fine-tuning, and retrieval-augmented LLM paradigms.

#### 4.1.3 Evaluation Metrics

We evaluate recommendation performance using two standard metrics widely adopted in cold-start tasks: Recall@$k$ and NDCG@$k$, following prior work (Hou et al., 2022; Wei et al., 2021; Liang et al., 2025). All results are reported at $k = 10$.

#### 4.1.4 Implementation Details

ColdRAG and all training-free baselines are implemented using two LLM backbones: *gpt-4o-mini*[2] and *Qwen2.5-32b-instruct*[3] ("GPT" and "Qwen" in Table 2 and Table 3). Using two distinct LLM backbones verifies that ColdRAG's effectiveness generalizes beyond a single architecture. We set the edge scoring threshold to $\lambda = 7$, the candidate pool size to $\theta_{\text{pool}} = 300$, and the final candidate set size to $\theta_{\text{top}} = 100$, consistent with Section 3. All experiments are repeated five times, and average results are reported for stability and reproducibility. Additional implementation details appear in Appendix B.

### 4.2 Overall Performance (RQ1)

Table 2 reports the performance of ColdRAG against *training-free* baselines on the *Games*, *Toys*, and *Office* datasets using two LLM backbones. ColdRAG consistently achieves the best results across all metrics and domains. ColdRAG improves Recall@10 by 39.73%, 26.76%, and 104.76% with GPT, and by 78.22%, 24.62%, and 130.99% with Qwen on *Games*, *Toys*, and *Office*, respectively. The larger gains in Recall@10 compared to NDCG@10 indicate that ColdRAG is particularly effective at retrieving relevant candidates into the top set while maintaining strong ranking quality. Among *training-free* baselines, LLM and LLM-Rank variants perform moderately but rely on predefined candidate lists, which restrict contextual exploration. KALM4Rec partially alleviates this limitation through keyword-based retrieval, yet remains constrained to shallow matching. In contrast, ColdRAG enables adaptive, multi-hop traversal over semantically linked concepts in a dynamically constructed knowledge graph, allowing it to retrieve more candidates and achieve higher accuracy.

Table 3 further compares ColdRAG with *training-based* cold-start models. ColdRAG consistently outperforms all dedicated training-based baselines, including TDRO, which represents the strongest model optimized through supervised learning objectives. This result holds across all domains and for both backbones, indicating that even carefully trained models struggle to generalize under item sparsity and distribution shift. In contrast, ColdRAG achieves superior performance without domain-specific parameter training, relying instead on retrieval-augmented generation with dynamically constructed knowledge graph.

Overall, results across both tables show that ColdRAG not only surpasses existing training-free approaches, but also outperforms established training-based cold-start models, highlighting the effectiveness of dynamic knowledge graph retrieval combined with LLM reasoning for robust item cold-start recommendation.

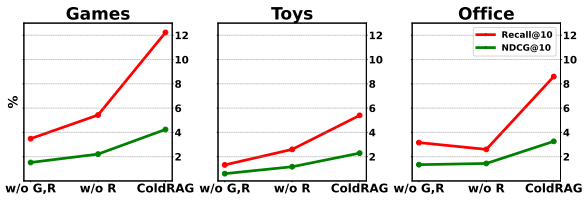


Figure 2: Performance comparison of ColdRAG variants across three domains using GPT, showing that both core modules (G and R) add performance gains.

### 4.3 Ablation Study (RQ2 & RQ3)

To examine the impact of ColdRAG's core components, we compare three settings: **w/o G,R**, a plain LLM without dynamic knowledge graph construction (**G**) or adaptive candidate retrieval (**R**); **w/o R**, a variant that includes G but replaces R with embedding-similarity top-k matching; and the full ColdRAG model combining both modules. As shown in Figure 2, performance improves steadily from **w/o G,R** to **ColdRAG** across most domains. These results indicate that dynamic knowledge graph construction (**G**) provides structured semantic grounding, while adaptive candidate retrieval (**R**) introduces goal-directed exploration over related entities, together yielding consistent gains as each module is added. In the *Office* domain, **w/o R** performs slightly worse than **w/o G,R**, indicating that unfiltered knowledge can introduce noise when metadata is sparse. However, the full model restores performance by selectively refining relevant information through reasoning. Overall, the two modules are complementary: knowledge grounding provides semantic depth, and retrieval ensures

[2] https://platform.openai.com/docs/models/gpt-4o-mini

[3] https://huggingface.co/Qwen/Qwen2.5-32B-Instruct

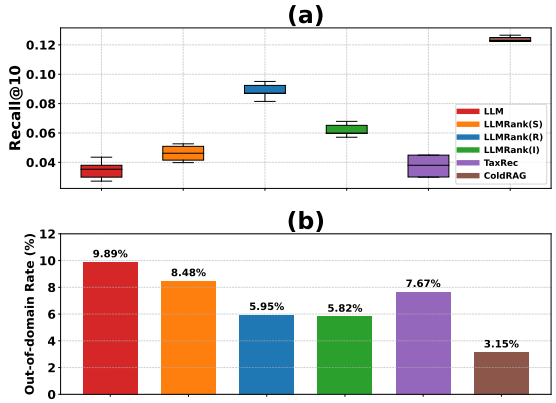


Figure 3: (a) Recall@10 box plots over five runs for training-free baselines. (b) Out-of-domain generation rates, both evaluated on the *Games* dataset with GPT.

relevance. Their combination drives ColdRAG's superior cold-start recommendation performance.

### 4.4 Analysis on Stability and Hallucination (RQ4 & RQ5)

LLM-based recommenders often suffer from generation inconsistency and hallucination, producing unstable or out-of-domain outputs. We evaluate ColdRAG on these aspects using five independent runs on the *Games* dataset. As shown in Figure 3 (a), ColdRAG achieves high average performance and low variance in Recall@10, demonstrating stable, reproducible generation compared to other training-free baselines. This robustness stems from structured retrieval and reasoning, providing consistent semantic grounding rather than relying on prompt randomness. We also measure hallucination rates, shown in Figure 3 (b), defined as the proportion of generated items not present in the dataset. While other LLM-based models, including LLM and LLMRank variants, exhibit 5–10% out-of-domain outputs even with predefined candidate lists, ColdRAG reduces this rate to 3.15%. This improvement shows that knowledge-grounded retrieval helps the model construct and reason over a semantic graph, enabling it to retrieve contextually valid items and constrain generation within the domain. In summary, beyond achieving superior recommendation performance, ColdRAG also exhibits robust stability and minimal hallucination, which are essential for practical and trustworthy recommender systems.

### 4.5 Hyperparameter Analysis

We analyze the impact of the edge scoring threshold $\lambda$, which controls how strictly ColdRAG filters edges during Query-aware Multi-hop Reasoning. As shown in Figure 4, ColdRAG achieves the best performance when $\lambda = 7$. A smaller threshold allows irrelevant edges to remain, while an excessively large threshold overly constrains traversal and misses useful nodes. This result indicates that a moderate threshold effectively balances relevance and diversity in the retrieved candidates, yielding the most robust overall performance.

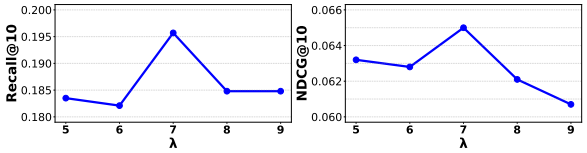


Figure 4: Effect of the edge scoring threshold $\lambda$ on ColdRAG's Recall@10 and NDCG@10 on the *Games* dataset.

## 5 Discussion

To provide deeper insight into ColdRAG's practical behavior beyond aggregate performance, we further discuss its adaptability to cold-start scenarios, its position among recommender system paradigms, and the structural properties of the constructed knowledge graph.

### 5.1 Cold-start Adaptability

ColdRAG is inherently suitable for item cold-start scenarios, as illustrated in Figure 5. When a new item appears with only metadata, the framework immediately generates its item profile, extracts entities and relations, and integrates them into the existing knowledge graph through the same pipeline used for prior items. This enables the new item to connect with semantically related concepts, such as genres, features, or characters, and to participate in the graph's reasoning and retrieval processes without requiring historical interactions or retraining.

Importantly, this integration is performed online and does not depend on periodic model updates or offline graph reconstruction, allowing ColdRAG to adapt continuously as the item catalog evolves. As a result, newly introduced items can be recommended as soon as their metadata becomes available, while still benefiting from the structured semantic context accumulated from existing items. Such seamless and incremental incorporation makes ColdRAG particularly robust in

dynamic environments where new items are frequently introduced.

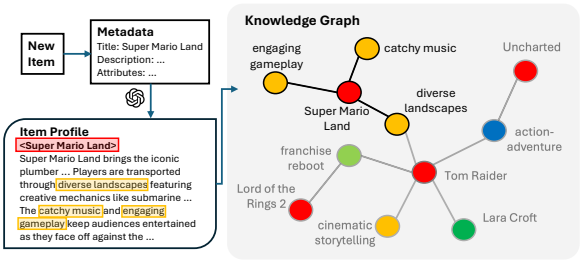


Figure 5: Illustration of ColdRAG's adaptability to item cold-start scenario.

## 5.2 Comparison of Recommender Paradigms

Table 4 situates ColdRAG within the broader landscape of recommender system paradigms. Traditional collaborative filtering (CF) and content-based (CB) methods address cold-start only partially and lack zero-shot capability, while hybrid models (CF+CB) still depend on interaction data to function effectively. Prompt-only LLM recommenders enable zero-shot inference but suffer from hallucination and weak grounding, as they rely solely on parametric knowledge. RAG-based LLMs improve factual grounding through external retrieval, yet typically operate over shallow or unstructured evidence and do not support relational, multi-hop reasoning.

ColdRAG uniquely combines dynamic external grounding, zero-shot usability, and multi-hop reasoning over structured knowledge, allowing it to handle item cold-start scenarios without retraining or predefined candidate lists. As summarized in Table 4, ColdRAG is the only framework that simultaneously satisfies all four key capabilities, highlighting its distinct position among existing recommender system categories.

Table 4: Comparison of recommender paradigms across four key capabilities: handling cold-start, grounding in external evidence, zero-shot usability, and multi-hop reasoning over structured knowledge. (* denotes static metadata rather than retrieved evidence.)

| | Cold-start | Grounding | Zero-shot | Multi-hop |
|---|---|---|---|---|
| CF | ✗ | ✗ | ✗ | ✗ |
| CB | ✓ | ✓* | ✗ | ✗ |
| CF + CB | ✓ | ✓ | ✗ | ✗ |
| Prompt-only LLM | ✓ | ✗ | ✓ | ✗ |
| RAG-based LLM | ✓ | ✓ | ✓ | ✗ |
| **ColdRAG (ours)** | ✓ | ✓ | ✓ | ✓ |

## 5.3 Structure of Knowledge Graph

We analyze the structure of the KG constructed using the *Games* dataset with *gpt-4o-mini*, focusing on the distribution of entity types and their relations, as shown in Figure 6 (a). The KG is primarily composed of *item* nodes (50%) and *feature* nodes (26%), while other entities such as *target user* (9%), *etc* (6%), *setting* (5%), and *genre* (4%) provide complementary semantic context. This distribution shows that the KG is centered around items, with non-item entities describing and explaining their properties. The heatmap in Figure 6 (b) reveals dense connections between items and features (15,102 edges) and between items and target users (4,928 edges), indicating that the graph effectively captures item attributes and user-related semantics. Additional links to genre and setting nodes further enrich contextual diversity, enabling nuanced multi-hop reasoning. Overall, the KG exhibits a dense yet interpretable structure that supports ColdRAG's retrieval and reasoning processes.

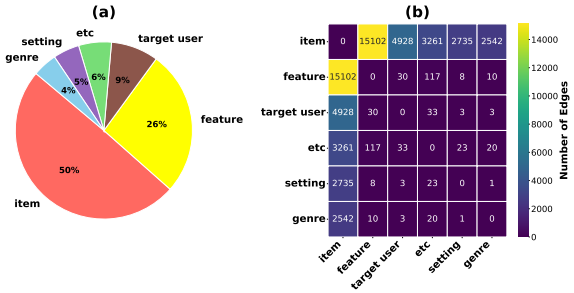


Figure 6: (a) Pie chart of the distribution of entity (node) types in the KG. (b) Heatmap of the number of relations (edges) between the entity types.

## 6 Conclusion

We presented ColdRAG, a retrieval-augmented generation framework for item cold-start recommendation. ColdRAG dynamically builds a knowledge graph from sparse metadata and performs LLM-guided multi-hop reasoning to adaptively retrieve candidate items aligned with user preferences, without relying on pre-built candidate lists. This design enables accurate and stable recommendations, making ColdRAG a step toward practical cold-start recommendation for real-world scenarios. Experiments across multiple domains show that ColdRAG consistently outperforms both training-based and training-free baselines. Moreover, its dynamic retrieval and reasoning pipeline improves robustness and reliability under evolving item catalogs.

## Limitations

While ColdRAG demonstrates superior performance, it faces several practical constraints. First, its reliance on repeated LLM queries during knowledge graph construction and multi-hop reasoning introduces notable computational cost and latency, posing challenges for large-scale or real-time deployment. Moreover, although ColdRAG can operate with both open- and closed-source LLMs, reliance on closed-sourced models such as the GPT series can make reproduction costly and less consistent across environments. Another limitation lies in ColdRAG's limited adaptability. Several key hyperparameters, such as edge scoring thresholds and candidate pool sizes, are manually set and static across domains. This rigidity may constrain performance under varying data distributions or interaction sparsity. A more adaptive, agentic framework could dynamically adjust these parameters and query strategies, improving both efficiency and generalization in diverse real-world settings.

## References


Manal A Alshehri and Xiangliang Zhang. 2022. Generative adversarial zero-shot learning for cold-start news recommendation. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pages 26–36.

Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 34(8):3549–3568.

Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 585–593.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. Large language models are zero-shot rankers for recommender systems. In *European Conference on Information Retrieval*, pages 364–381. Springer.

Feiran Huang, Zefan Wang, Xiao Huang, Yufeng Qian, Zhetao Li, and Hao Chen. 2023. Aligning distillation for cold-start item recommendation. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 1147–1157.

Umair Javed, Kamran Shaukat, Ibrahim A Hameed, Farhat Iqbal, Talha Mahboob Alam, and Suhuai Luo. 2021. A review of content-based and context-based recommendation systems. *International Journal of Emerging Technologies in Learning (iJET)*, 16(3):274–306.

Yangqin Jiang, Yuhao Yang, Lianghao Xia, and Chao Huang. 2024. Diffkg: Knowledge graph diffusion model for recommendation. In *Proceedings of the 17th ACM international conference on web search and data mining*, pages 313–321.

Hai-Dang Kieu, Minh-Duc Nguyen, Thanh-Son Nguyen, and Dung D Le. 2025. Keyword-driven retrieval-augmented large language models for cold-start user recommendations. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 2717–2721.

Jingjing Li, Mengmeng Jing, Ke Lu, Lei Zhu, Yang Yang, and Zi Huang. 2019. From zero-shot learning to cold-start recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4189–4196.

Yueqing Liang, Liangwei Yang, Chen Wang, Xiongxiao Xu, S Yu Philip, and Kai Shu. 2025. Taxonomy-guided zero-shot recommendations with llms. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1520–1530.

Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Hao Zhang, Yong Liu, Chuhan Wu, Xiangyang Li, Chenxu Zhu, and 1 others. 2025. How can recommender systems benefit from large language models: A survey. *ACM Transactions on Information Systems*, 43(2):1–47.

Xinyu Lin, Wenjie Wang, Jujia Zhao, Yongqi Li, Fuli Feng, and Tat-Seng Chua. 2024. Temporally and distributionally robust optimization for cold-start recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8750–8758.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 188–197.

Scott Sanner, Krisztian Balog, Filip Radlinski, Ben Wedin, and Lucas Dixon. 2023. Large language models are competitive near cold-start recommenders for language-and item-based preferences. In *Proceedings of the 17th ACM conference on recommender systems*, pages 890–896.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450.

Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019a. Multi-task feature learning for knowledge graph enhanced recommendation. In *The world wide web conference*, pages 2000–2010.

Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019b. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 950–958.

Yinwei Wei, Xiang Wang, Qi Li, Liqiang Nie, Yan Li, Xuanping Li, and Tat-Seng Chua. 2021. Contrastive learning for cold-start recommendation. In *Proceedings of the 29th ACM international conference on multimedia*, pages 5382–5390.

Riya Widayanti, Mochamad Heru Riza Chakim, Chandra Lukita, Untung Rahardja, and Ninda Lutfiani. 2023. Improving recommender systems using hybrid techniques of collaborative filtering and content-based filtering. *Journal of Applied Data Sciences*, 4(3):289–302.

Xuansheng Wu, Huachi Zhou, Yucheng Shi, Wenlin Yao, Xiao Huang, and Ninghao Liu. 2024. Could small language models serve as recommenders? towards data-centric cold-start recommendation. In *Proceedings of the ACM Web Conference 2024*, pages 3566–3575.

Weizhi Zhang, Yuanchen Bei, Liangwei Yang, Henry Peng Zou, Peilin Zhou, Aiwei Liu, Yinghui Li, Hao Chen, Jianling Wang, Yu Wang, and 1 others. 2025. Cold-start recommendation towards the era of large language models (llms): A comprehensive survey and roadmap. *arXiv preprint arXiv:2501.01945*.

# A Prompt Templates

We present the prompt templates used in the four core modules of ColdRAG. Each prompt is designed to guide the LLM through a distinct stage of the pipeline, ensuring consistent and interpretable behavior.

## A.1 Item Profile Generation

This prompt directs the LLM to create a concise, fluent item profile using the title, metadata, and reviews, enriching sparse information with its pretrained knowledge when necessary.

```
You are an expert in gaming and consumer electronics.

For each product, write a short, engaging, and informative description.
If the given information is insufficient to describe the item,
enrich the description using your own knowledge about the product or its context.

Example:
"### Item Title: The Legend of Zelda: Breath of the Wild (Nintendo Switch) ###
One of the most acclaimed open-world games ever made ... "

Now, follow the given example format and write a description for the following item:
"Item Title: {item title},
Category: {categories},
Description: {item description},
Reviews: {item reviews}"
```

Figure 7: Example prompt for Item Profile Generation.

## A.2 Dynamic Knowledge Graph Construction

This prompt instructs the LLM to extract entities and relations from the generated item profile, producing a structured and interpretable knowledge graph centered around the item.

```
You are an entity and relationship extraction specialist.
Given a product description and reviews,
identify entities and relations that describe the item.

1. Extract entities with:
  - entity_name: If "item", use exact full title from the line `### Item Title: ... ###`.
    For other types (genre, setting, feature, target user, etc.), extract distinct phrases.
  - entity_type: ["item", "genre", "setting", "feature", "target user", "etc"].
  - entity_description: A concise explanation of its meaning or role.

2. Extract relationships:
  - Every non-item entity must connect to item entity.
  - Include: source_entity, target_entity, relationship_description.

Input text: {text}
```

Figure 8: Example prompt for Entity and Relation Extraction.

## A.3 Adaptive Candidate Retrieval over KG

This prompt enables the LLM to evaluate graph edges using the user's interaction history and iteratively expand the reasoning frontier to identify semantically relevant candidate items.

```
You are helping with multi-hop reasoning over a knowledge graph.
The following is the user's past interaction history: {history}

Below are edges from the current reasoning frontier, each connecting two entities.
For each edge, assess how strongly it relates to the user's past interests.
Score each edge from 0 (completely irrelevant) to 10 (highly relevant) based on
semantic similarity to the user's interaction history.

Here are the edges: {edges}
```

Figure 9: Example prompt for Adaptive Candidate Retrieval.

## A.4 Retrieval-augmented Generation

This prompt guides the LLM to rank the retrieved candidate items and produce the final top-$k$ recommendations in a dataset-consistent format.

```
You are a recommendation assistant.
Given the following item interactions: {history}
And the following candidate item list: {candidate_list}

Recommend the top {k} items that the user would most likely interact with next,
based on the user's past interactions.

Only output the final ranked list in this strict format:
"1. <item title>\n2. <item title>\n...\n(up to {k} items)\n\n"
```

Figure 10: Example prompt for Retrieval-augmented Generation.

## B Additional Implementation Details

For the GPT setting, we use *gpt-4o-mini* accessed through the Azure OpenAI API. Entity and relation embeddings are encoded using OpenAI's *text-embedding-3-small* model, with all embeddings indexed in *FAISS* for approximate nearest-neighbor retrieval. For the Qwen setting, we employ *qwen2.5-32b-instruct* served via the *vLLM*[4] backend, paired with the *bge-m3*[5] embedding model for semantic representation. Both configurations follow identical hyperparameters and retrieval settings to ensure a fair comparison across LLM backbones. These results confirm that ColdRAG's performance is consistent across different LLM architectures, demonstrating its architecture-agnostic robustness.

[4] https://github.com/vllm-project/vllm
[5] https://huggingface.co/BAAI/bge-m3