

When Graphs Whisper: Graph Structure Serialization for Embedding Alignment in Cold-Start Recommendation

Yujing Zhang^{1*}, Huachi Zhou^{1*}, Ruochen Liu², Qinggang Zhang^{3†},
Luyao Zhuang¹, Hao Chen⁴, Zijin Hong¹, Chang Yang¹, Xiao Huang¹

¹The Hong Kong Polytechnic University, ²Central South University,
³Jilin University, ⁴City University of Macau

{yu-jing.zhang, huachi.zhou}@connect.polyu.hk, qinggangzhang@jlu.edu.cn, xiao.huang@polyu.edu.hk

Abstract

Cold-start item recommendation remains challenging as new items lack interaction signals beyond textual descriptions. Existing methods either fine-tune large language models (LLMs) on descriptions or map item content into collaborative spaces learned from user-warm item interactions. However, text-only prompts struggle to capture higher-order graph structures, while distributional differences between behavior and content embeddings weaken alignment based solely on predicted probabilities. To address these issues, we propose **Graph Structure Serialization for Embedding Alignment (GraphSEA)**, an LLM-based framework inspired by the observation that graphs contain sequential structures similar to language and can be serialized as inputs for LLMs. GraphSEA first serializes interaction graphs into structured token sequences for an LLM to learn behavior embeddings that encode collaborative patterns, while the LLM separately generates content embeddings from item descriptions. An embedding alignment module then aligns behavior and content embeddings across multiple facets, including interaction probability, preference order, and item distinction. Extensive experiments demonstrate that GraphSEA improves recommendation performance across warm and cold recommendation tasks, while achieving higher efficiency than LLM-based methods.

1 Introduction

In real-world recommender systems, a large number of newly listed products are introduced to platforms every day (Luo et al., 2025; Zhang et al., 2025b). Cold-start item recommendation refers to the task of effectively recommending these items, which is crucial for ensuring both the quality and freshness of recommendation systems (Zhang et al., 2025b; Sethi and Mehrotra, 2021). While cold

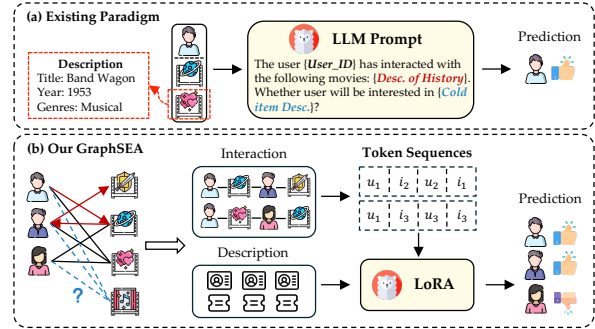


Figure 1: Comparison between (a) **Existing Paradigm** and (b) **Our GraphSEA**. Unlike existing methods that only leverage first-order interactions through textual prompts, GraphSEA transforms the graph into token sequences to model higher-order interactions while learning content information from textual descriptions.

items lack user interactions, the rich collaborative information from user-warm item interactions provides valuable guidance for cold item recommendation (Wang et al., 2024c).

Traditional approaches (Xu et al., 2022; Shi et al., 2019) address this problem by learning a mapping function. They first generate content embeddings from item features, e.g., textual descriptions, and then map content embeddings into the behavior embedding space learned from warm items with interactions (Chae et al., 2019; Sun et al., 2020). However, these methods are often constrained by the limited reasoning capacity of their mapping modules. Recent methods (Liu et al., 2026; Wang et al., 2024a) leverage large language models (LLMs) to model user-item interactions. ColdLLM (Huang et al., 2025) fine-tunes LLMs to extract behavior embeddings from user-interacted item descriptions, which are then concatenated with content embeddings to predict user-item interactions. FilterLLM (Liu et al., 2025) first extends the LLM vocabulary with user behavior and then maximizes the mutual information between behavior and content embeddings via extensive training.

*Equal contribution.

†Corresponding author.

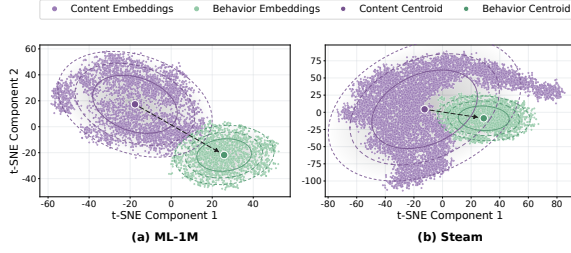


Figure 2: Visualization of **Content Embeddings** and **Behavior Embeddings** on Two Datasets. Content embeddings are derived from textual descriptions, while behavior embeddings emerge from collaborative interactions, showing clear distributional divergence.

Despite these advances, existing LLM-based methods still face two major issues: (i) Input constraints of LLMs. As shown in Figure 1, existing methods mainly rely on textual prompts and thus capture only first-order interactions, overlooking higher-order graph structures that encode richer behavioral similarities. Moreover, converting complex user-item interactions into plain text sequences may lose critical topological information (Liu et al., 2023a). (ii) Distributional difference in embedding alignment. Existing methods primarily align the interaction probabilities predicted by content and behavior embeddings. However, behavior embeddings are learned from historical interactions, whereas cold-item content embeddings are derived from textual descriptions, making them difficult to align in a unified space (Li and Kong, 2025; Zhu et al., 2026). As shown in Figure 2, behavior embeddings form compact clusters, while content embeddings are more scattered due to the diverse and noisy nature of textual descriptions.

These limitations suggest that cold-start recommendation requires a representation form that encodes collaborative graph structures for LLMs. Although graphs are not natural language (Feng et al., 2026; Zhang et al., 2025a), user-item interaction graphs naturally contain sequential structures similar to language: connected users and items form paths, and these paths can be organized as token sequences. Through structure serialization, the collaborative signals hidden in graphs can be exposed to LLMs as structured inputs. Moreover, deriving both behavior embeddings from serialized graph structures and content embeddings from item descriptions with LLMs can make the two representations more compatible, thereby facilitating embedding alignment. Motivated by these insights, we propose GraphSEA, an LLM-based framework for

cold-start item recommendation.

GraphSEA first introduces an LLM-based embedding generation module that serializes user-item interaction graphs into structured token sequences. This graph structure serialization preserves the connectivity between users and items, enabling LLMs to model complex topological structures and higher-order interactions through fine-tuning. In this module, we also leverage pre-trained LLMs to generate content embeddings from item descriptions. Finally, we design an embedding alignment module that compares behavior and content embedding distributions and aligns them through three facets: interaction probability, preference order, and item distinction. Our main contributions are as follows:

- We propose GraphSEA, a novel LLM-based framework for cold-start item recommendation via graph structure serialization and behavior-content embedding alignment.
- We introduce a graph structure serialization that converts interaction graphs into structured token sequences, capturing rich collaborative information essential for cold item learning.
- We design a cold embedding alignment module that aligns behavior and content embedding distributions by considering three facets.
- Extensive experiments on two datasets demonstrate that GraphSEA achieves significant improvements in effectiveness and efficiency.

2 Related Work

Cold-Start Recommendation. The Cold-Start Problem refers to the challenge of providing recommendations when no historical interaction data is available (Jangid and Kumar, 2025; Li et al., 2025; Chen et al., 2026). While this challenge appears in various scenarios (Fu et al., 2019; Wang et al., 2020; Van den Oord et al., 2013), it is most commonly encountered in cold items (Bai et al., 2023; Wei et al., 2017). Traditional methods can be broadly divided into two categories. Dropout-based models (Volkovs et al., 2017; Du et al., 2020) introduce the absence or substitution of behavior embeddings during training, allowing the model to effectively learn embeddings for cold items under limited collaborative information. Generative-based models (Pan et al., 2019; Zhu et al., 2021b; Chang et al., 2026) learn the mapping between content

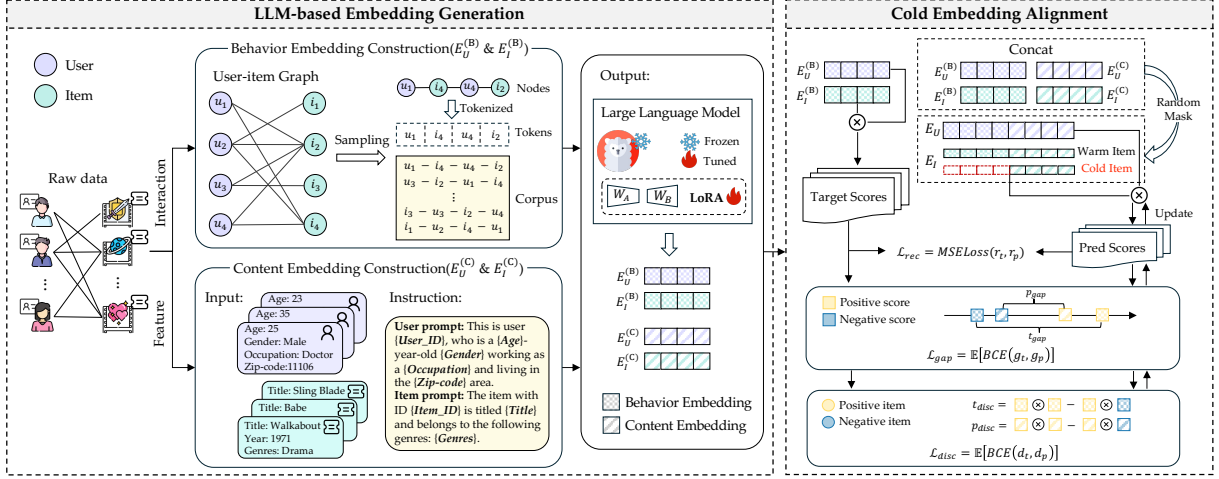


Figure 3: The framework of **GraphSEA**. (i) **LLM-Based Embedding Generation** (left) transforms the user-item interaction graph into structured token sequences to generate behavior embeddings, while leveraging LLMs to generate content embeddings from textual descriptions; (ii) **Cold Embedding Alignment** (right) aligns behavior and content embeddings through multiple facets, including interaction probability, preference order, and item distinction.

embeddings derived from cold item descriptions and behavior embeddings based on warm items.

LLMs for recommendation. The integration of LLMs into recommender systems has emerged as a transformative paradigm (Zhang et al., 2024; Chang et al., 2024). Addressing the specific challenges of cold-start scenarios, researchers have developed targeted approaches that leverage LLMs’ sophisticated reasoning capabilities (Sakurai et al., 2025; Rong et al., 2024; Uruqi et al., 2025). Wang et al. (Wang et al., 2024a) utilize LLMs as data augmenters to bridge knowledge gaps for cold items by inferring user preferences based on textual descriptions of behaviors and items. ColdLLM (Huang et al., 2025) introduces an LLM simulator framework that simulates user interactions for cold items while managing computational complexity through a coupled funnel approach that filters billions of users to hundreds for efficient LLM processing. FilterLLM (Liu et al., 2025) proposes a “Text-to-Distribution” paradigm that predicts interaction probability distributions for entire user sets in a single inference, incorporating efficient user-vocabulary structures for billion-scale deployment.

3 Preliminaries

Notations. Let $\mathcal{U} = \{u_1, \dots, u_N\}$ denote the set of N users and $\mathcal{I} = \{i_1, \dots, i_M\}$ denote the set of M items, where $\mathcal{I}$ is partitioned into warm items $\mathcal{I}_w = \{i_1, \dots, i_K\}$, which have interaction history, and cold items $\mathcal{I}_c = \{i_{K+1}, \dots, i_M\}$, which have no interaction history. The user-item interaction

graph is represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{D})$, where $\mathcal{V} = \mathcal{U} \cup \mathcal{I}_w$ denotes the node set consisting of the user set $\mathcal{U}$ and the warm-item set $\mathcal{I}_w$, $\mathcal{E}$ represents the interactions between users and warm items, and $\mathcal{D}$ contains the textual descriptions used for content embedding, including descriptions for all items in $\mathcal{I}$. The interaction information can also be encoded as a matrix $\mathcal{R} \in \mathbb{R}^{N \times K}$, where $|\mathcal{R}|$ is the total number of interactions. Each element $r_{ui} \in \mathcal{R}$ indicates the existence of an interaction between user u and warm item i .

Problem Definition. In this paper, we focus on the cold item recommendation task based on collaborative filtering, aiming to predict the preference score $\hat{y}_{ui}$ of user $u \in \mathcal{U}$ for item $i \in \mathcal{I}$. Traditional methods learn user preferences from historical interaction data by decomposing the user-item interaction matrix $\mathcal{R} \in \mathbb{R}^{N \times K}$ to obtain user embeddings $E_U^{(B)} \in \mathbb{R}^{N \times d}$ and item embeddings $E_{I_w}^{(B)} \in \mathbb{R}^{K \times d}$, where d is the embedding dimension. For all items $i \in \mathcal{I}$, their textual descriptions are utilized to construct content embeddings $E_I^{(C)} \in \mathbb{R}^{M \times d}$. The prediction score is defined as:

$$\hat{y}_{ui} = f(E_U^{(B)}, E_{I_w}^{(B)}, E_I^{(C)}), \quad (1)$$

where $f(\cdot)$ generates user-item interaction probability based on the input embeddings.

4 Methodology

In this section, we propose a novel framework called GraphSEA. As shown in Figure 3, GraphSEA consists of two modules: (i) **LLM-Based**

Embedding Generation: Use LLMs to generate behavior embeddings based on interaction data and content embeddings from textual descriptions. Behavior embeddings are derived through a graph structure serialization that encodes user-item interaction graphs as sequential tokens, preserving node relationships and higher-order interactions. Content embeddings are generated by utilizing pre-trained LLMs to process item descriptions and extract meaningful representations. **(ii) Cold Embedding Alignment:** A customized alignment learning mechanism aligns behavior and content embeddings in multiple facets, including interaction probability, preference order, and item distinction, through tailored loss functions. The whole process of our GraphSEA can be seen in Appendix A.1.1.

4.1 LLM-Based Embedding Generation

Behavior Embedding Construction. To enable the LLM to understand the structural information within the user-item graph, we design a graph structure serialization where each node is represented as a token, and connecting paths between nodes are transformed into token sequences. We employ a random walk strategy to sample paths on the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{D})$. Specifically, given the current node $v_i \in \mathcal{V}$ with neighbor set $\Gamma(v_i)$, the probability P of selecting the next node v_{i+1} is:

$$P(v_{i+1}|v_i) = \frac{1}{|\Gamma(v_i)|}. \quad (2)$$

This sampling process generates a structured sequence corpus $\mathcal{S} = \{s_1, \dots, s_k\}$ that encodes collaborative information. For instance, in Figure 3, u_1 is represented as a token, and the path $s_i = \{u_1, i_4, u_4, i_2\}$ forms a sequence. Since the empirical transition probabilities estimated from sampled walks converge to the true graph transition probabilities as the number of walks increases, graph structure serialization can retain high-order behavioral relationships in the user-item interaction graph. The formal statement and proof are provided in Appendix A.1.2.

Based on the generated graph corpus, we fine-tune the LLM by maximizing the likelihood of next token prediction over the sequence corpus. Formally, the fine-tuning objective is defined as:

$$\mathcal{L}_{\text{pre}} = - \sum_{i=1}^{|\mathcal{S}|} \sum_{j=1}^{|s_i|} \log P(s_{i,j} | s_{i,<j}), \quad (3)$$

where $s_{i,j}$ represents the j -th token of the i -th sequence, $s_{i,<j}$ represents its preceding context.

Through this objective, the LLM learns to encode the collaborative information that is inherently embedded within the sequences.

After fine-tuning, the hidden representations from the fine-tuned model are aggregated to generate behavior embeddings. Specifically, for each serialized graph sequence $s_i = (s_{i,1}, \dots, s_{i,|s_i|})$, we obtain the final-layer hidden states:

$$\mathbf{H}_i = \text{LLM}_\theta(s_i) = [\mathbf{h}_{i,1}, \mathbf{h}_{i,2}, \dots, \mathbf{h}_{i,|s_i|}], \quad (4)$$

where $\mathbf{h}_{i,j}$ denotes the hidden representation at the j -th position of sequence s_i . Under the causal attention mask, $\mathbf{h}_{i,j}$ is computed from the prefix $\{s_{i,1}, \dots, s_{i,j}\}$, thereby encoding both the current token and its preceding structural context.

To comprehensively encode higher-order collaborative information, we aggregate the contextualized representations of each token over the entire serialized corpus:

$$\mathbf{h}_v = \sum_{s_i \in \mathcal{S}} \sum_{j=1}^{|s_i|} \mathbb{I}(s_{i,j} = v) \mathbf{h}_{i,j}, \quad (5)$$

where $\mathbb{I}(\cdot)$ is an indicator function that equals 1 if the condition holds and 0 otherwise. This aggregation collects all contextualized occurrences of token v in the corpus $\mathcal{S}$. Therefore, the behavior embeddings of users and warm items can be expressed as:

$$E_U^{(B)} = \{\mathbf{h}_v | v \in \mathcal{U}\}, E_{I_w}^{(B)} = \{\mathbf{h}_v | v \in \mathcal{I}_w\}. \quad (6)$$

Content Embedding Generation. Traditional word embedding methods (Johnson et al., 2024; Qiu et al., 2024) are limited by their model capacity and struggle to capture deep semantic relationships in textual data. Thus, we employ LLMs to generate high-quality content embeddings through designed prompt templates.

As illustrated in Figure 3, we convert structured user and item information into natural language descriptions using domain-specific prompts. This approach creates a unified textual format for LLM processing, resulting in a corpus $\mathcal{D} = \{T_{u_1}, \dots, T_{u_N}, T_{i_1}, \dots, T_{i_M}\}$ that encapsulates structured representations of users and items. We then perform domain-adaptive fine-tuning on this corpus to enable the pre-trained LLM to acquire deep understanding of recommendation-specific semantic information and domain knowledge.

Following this fine-tuning phase, the adapted model generates dense embeddings by aggregating

the token-level hidden states from the final Transformer layer using mean pooling. Specifically, the representation of each prompt is computed as the arithmetic mean of its constituent hidden states:

$$\mathbf{h}_k = \frac{1}{|x_k|} \sum_{t=1}^{|x_k|} \mathbf{h}_{k,t}, \quad (7)$$

where $\mathbf{h}_{k,t} \in \mathbb{R}^d$ denotes the d -dimensional hidden state at position t in sequence k . Therefore, user and item content embeddings are expressed as:

$$E_U^{(C)} = \{\mathbf{h}_k | k \in U\}, E_I^{(C)} = \{\mathbf{h}_k | k \in I\}. \quad (8)$$

The resulting representations encode local semantic details via hidden states and capture global relational patterns through mean aggregation.

4.2 Cold Embedding Alignment

Concatenation and Masking. We concatenate behavior embeddings and content embeddings along the feature dimension, which can be formulated as:

$$E_U = E_U^{(B)} \oplus E_U^{(C)}, E_{I_w} = E_{I_w}^{(B)} \oplus E_{I_w}^{(C)}, \quad (9)$$

where E_U and E_{I_w} denote the concatenated embeddings of users and warm items, respectively. Compared to more complex feature fusion methods, such as attention mechanisms (Vaswani et al., 2017), this simple concatenation operation offers extremely high computational efficiency.

To simulate the cold item scenario in recommender systems, we propose an innovative random masking strategy that constructs cold item scenarios by selectively masking a portion of the behavior embeddings. This process can be formulated as:

$$E_{I_w}^{(\bar{B})} = M_{ij} \odot E_{I_w}^{(B)}, \quad (10)$$

where $\odot$ denotes the Hadamard product. Each mask variable M_i follows a Bernoulli distribution with parameter $(1 - r)$ and is replicated across dimensions $j \in 1, 2, \dots, d$ to form M_{ij} . When $M_i = 1$, the corresponding embedding vector (row i of $E_{I_w}^{(B)}$) is retained; when $M_i = 0$, it is masked.

After the masking operation, the resulting concatenated embeddings of items are expressed as:

$$E_{\bar{I}_w} = E_{I_w}^{(\bar{B})} \oplus E_{I_w}^{(C)}. \quad (11)$$

It is worth emphasizing that the masking operation is dynamically applied to warm items during training. Specifically, for each warm item, whether

to apply masking is randomly determined in each training batch, preventing the model from overfitting to specific instances and improving its generalization and robustness. For cold items $\mathcal{I}_c$, behavior embeddings $E^{(B)} I_c$ are not constructed and are therefore treated as fully masked, while only the corresponding content embeddings $E_{I_c}^{(C)}$ are retained for representation learning.

Alignment Objectives. In the alignment process, we design several alignment objectives to guide the model in learning effective representations of cold items from behavior embeddings by comparing differences between behavior embeddings and concatenated embeddings.

Firstly, we employ a Score Reconstruction Loss based on mean squared error (MSE) to directly measure the difference between the predicted scores and the target scores. This loss encourages the predicted scores to closely approximate the target interactions, providing a stable numerical foundation for subsequent optimization. The loss function is defined as follows:

$$\mathcal{L}_{\text{rec}} = \frac{1}{|\mathcal{N}|} \sum_{i=1}^{|\mathcal{N}|} (r_t - r_p)^2, \quad (12)$$

where $\mathcal{N}$ denotes the set of training instances used to compute the corresponding loss in the current batch, $r_t = E_U^{(B)} \cdot E_{I_w}^{(B)}$ is the target score derived from behavior embeddings of user-warm item interactions, and $r_p = f_U(E_U) \cdot f_I(E_{\bar{I}_w})$ is the predicted score derived from concatenated embeddings. Here, $f_U(\cdot)$ and $f_I(\cdot)$ are the mapping functions for user and item embeddings, respectively.

Secondly, we propose a Preference Gap Loss to align the preference gap between positive and negative samples in the target and predicted score spaces. By comparing the score differences between positive and negative samples, the model learns to preserve the essential relative preference structure, rather than merely matching absolute score values. This design is particularly suitable for recommendation scenarios, where users usually care more about the relative order of items than their absolute scores. Specifically, the Preference Gap Loss is defined as follows:

$$\mathcal{L}_{\text{gap}} = -\frac{1}{|\mathcal{N}|} \sum_{i=1}^{|\mathcal{N}|} (g_t^{(i)} \ln g_p^{(i)} + (1 - g_t^{(i)}) \ln(1 - g_p^{(i)})), \quad (13)$$

where $g_t = \sigma(r_{t,+} - r_{t,-})$ denotes the sigmoid-normalized gap between positive and negative target scores, and $g_p = \sigma(r_{p,+} - r_{p,-})$ denotes the

sigmoid-normalized gap between positive and negative predicted scores.

Finally, the Discriminative Consistency Loss maintains the discriminative structure of the embedding space by comparing the semantic relationships between positive and negative samples. This loss ensures that, even when concatenated embeddings are masked, the semantic distances between positive and negative samples remain reasonably distributed, thereby preserving their distinction. The loss function is defined as:

$$\mathcal{L}_{\text{disc}} = -\frac{1}{|\mathcal{N}|} \sum_{i=1}^{|\mathcal{N}|} (d_t^{(i)} \ln d_p^{(i)} + (1 - d_t^{(i)}) \ln(1 - d_p^{(i)})), \quad (14)$$

where d_t and d_p represent sigmoid-normalized differences between positive and negative embeddings in the target and predicted spaces, respectively. Formally, they are defined as:

$$\begin{aligned} d_t &= \sigma((E_{I,+}^{(B)})^T \cdot (E_{I,+}^{(B)} - E_{I,-}^{(B)})), \\ d_p &= \sigma((E_{\bar{I},+}^{(B)})^T \cdot (E_{\bar{I},+}^{(B)} - E_{\bar{I},-}^{(B)})). \end{aligned} \quad (15)$$

To balance different optimization objectives, we introduce two manually adjustable weight parameters to control the contribution of each auxiliary loss. The overall loss function is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rec}} + \alpha \cdot \mathcal{L}_{\text{gap}} + \beta \cdot \mathcal{L}_{\text{disc}}, \quad (16)$$

where α controls the contribution of $\mathcal{L}_{\text{gap}}$ and β controls the strength of $\mathcal{L}_{\text{disc}}$. This weighted strategy enables the model to balance multiple objectives, maintaining accuracy while preserving the preference gap and discriminative consistency.

5 Experiments

In this section, we present a comprehensive evaluation of our proposed method using two datasets, aiming to address the following research questions.

RQ1: Does GraphSEA outperform contemporary state-of-the-art cold-start recommendation models in both warm-item and cold-item scenarios?

RQ2: How does GraphSEA perform in terms of efficiency compared to other LLM-based baselines?

RQ3: How do different design choices affect the performance of GraphSEA, particularly regarding graph serialization and two custom losses in the alignment module? **RQ4:** How sensitive is GraphSEA to key hyperparameters, including the weighting factors α and β ? **RQ5:** How does the choice of LLM backbone affect the performance of GraphSEA? Additional experiments on sampling lengths,

sampling strategy, masking ratios, and embedding alignment differences are detailed in Appendix A.3.

5.1 Experiment Setup

Datasets. We conduct experiments on two datasets: MovieLens-1M (Harper and Konstan, 2015) with 6,040 users and 3,706 movies, and Steam (Sifa et al., 2015) with 43,727 users and 8,204 games. Following prior settings, we randomly designate 20% of items as cold items. The remaining 80% warm items are split into training, validation, and test sets with a ratio of 8:1:1. More dataset details are provided in Appendix A.2.1.

Baselines. We compare GraphSEA with representative baselines from three paradigms: (i) Dropout-based models: DropoutNet (Volkovs et al., 2017), MTPR (Du et al., 2020), Heater (Zhu et al., 2020), and CLCRec (Wei et al., 2021). (ii) Generative-based models: DeepMusic (Van den Oord et al., 2013), MetaEmb (Pan et al., 2019), GAR (Chen et al., 2022), and ALDI (Huang et al., 2023). (iii) LLM-based models: ColdLLM (Huang et al., 2025) and FilterLLM (Liu et al., 2025). More details of the baselines are provided in Appendix A.2.2.

Evaluation Metrics. We evaluate our proposed method and baselines on both warm and cold scenarios, adopting Precision@K and NDCG@K as primary metrics, where $k = 5, 10$. Precision@K evaluates the accuracy of the top- K recommendation set, while NDCG@K additionally accounts for ranking positions.

Implementation Setting. In our GraphSEA, the random walk sampling length is set to 4, and the masking ratio is fixed at 0.5. To ensure a fair comparison, we adopt LLaMA3.2-1B as the base LLM. Additional details regarding the implementation settings can be found in Appendix A.2.3.

5.2 Main Comparison (RQ1)

In the performance comparison experiments, we evaluated our GraphSEA against ten baselines under both warm and cold item scenarios. The experimental results are presented in Table 1.

As shown in the results, our method achieves the strongest overall performance across the two datasets and two evaluation scenarios. In particular, GraphSEA consistently delivers substantial improvements on cold item tasks, demonstrating its effectiveness in alleviating the cold-start problem. The improvement ranges from 0.49% to 9.37% on warm item tasks, and is even more significant on cold item tasks, reaching 7.17% to 42.49%. This

Model	ML-1M				Steam			
	PRE@5	PRE@10	NDCG@5	NDCG@10	PRE@5	PRE@10	NDCG@5	NDCG@10
<i>Warm Item Task</i>								
DropoutNet	0.2325	0.2130	0.2482	0.2347	0.1610	0.1142	0.3002	0.2993
MTPR	0.2108	0.1861	0.2207	0.2042	0.1610	0.1140	0.2873	0.2890
Heater	0.2369	0.2165	0.2544	0.2402	<u>0.1623</u>	<u>0.1150</u>	0.3015	0.3006
CLCRec	<u>0.2431</u>	<u>0.2215</u>	0.2543	0.2418	0.1477	0.1097	0.2794	0.2954
DeepMusic	0.2382	0.2179	0.2562	0.2427	0.1617	0.1133	0.3013	0.2991
MetaEmb	0.2357	0.2161	0.2551	0.2417	0.1623	0.1141	0.3032	0.2917
GAR	0.2392	0.2143	0.2521	0.2357	0.1590	0.1131	0.2967	0.2960
ALDI	0.2371	0.2179	0.2552	0.2426	0.1616	0.1123	0.3042	0.2973
ColdLLM	0.2088	0.1918	0.2210	0.2129	0.1477	0.1076	<u>0.3063</u>	0.3208
FilterLLM	0.1986	0.1830	0.2105	0.2027	0.1534	0.1131	0.3006	<u>0.3190</u>
Ours	0.2539	0.2229	0.2675	0.2459	0.1775	0.1255	0.3078	0.3067
%Improv.	↑4.44%	↑0.63%	↑4.41%	↑1.32%	↑9.37%	↑9.13%	↑0.49%	↓4.40%
<i>Cold Item Task</i>								
DropoutNet	0.0458	0.0493	0.0442	0.0491	0.0144	0.0112	0.0218	0.0221
MTPR	0.0510	0.0494	0.0519	0.0519	0.0078	0.0077	0.0093	0.0118
Heater	0.0654	0.0622	0.0667	0.0666	0.0090	0.0113	0.0101	0.0147
CLCRec	0.0414	0.0377	0.0435	0.0419	0.0159	0.0150	0.0206	0.0258
DeepMusic	0.0783	0.0828	0.0773	0.0836	0.0111	0.0151	0.0138	0.0206
MetaEmb	0.0176	0.0123	0.0225	0.0175	0.0175	0.0164	0.0182	0.0226
GAR	0.0723	0.0772	0.0701	0.0765	0.0024	0.0022	0.0036	0.0040
ALDI	0.0574	0.0491	0.0646	0.0592	0.0130	0.0132	0.0184	0.0228
ColdLLM	0.0867	0.0756	0.0873	0.0766	0.0245	0.0160	0.0233	0.0236
FilterLLM	<u>0.0934</u>	<u>0.0837</u>	<u>0.0928</u>	<u>0.0847</u>	<u>0.0281</u>	<u>0.0204</u>	<u>0.0273</u>	<u>0.0296</u>
Ours	0.1095	0.0897	0.1137	0.1022	0.0357	0.0250	0.0389	0.0398
%Improv.	↑17.24%	↑7.17%	↑22.52%	↑20.66%	↑27.05%	↑22.55%	↑42.49%	↑34.46%

Table 1: Comparison of **recommendation performance** for different baselines on **warm** and **cold item** tasks. “%Improv.” denotes the relative performance improvement compared to the best existing baseline method. The best result for each dataset is highlighted in **bold**, while the second-best result is indicated with an underline.

demonstrates that our method successfully combines the text understanding capabilities of LLMs with the collaborative filtering advantages of recommender systems, effectively balancing the performance for both warm and cold items. The larger gains in cold scenarios further indicate that the alignment effectively narrows the discrepancy between warm and cold item representations.

The results also show that LLM-based baselines achieve better performance than traditional methods in cold scenarios. For example, on the ML-1M dataset, FilterLLM achieves a PRE@5 score of 0.0934, exceeding most non-LLM baselines whose scores mainly fall between 0.04 and 0.07. This suggests that LLMs can better exploit textual descriptions to compensate for the lack of interaction signals in cold-start recommendation. Nevertheless, GraphSEA further improves upon these LLM-based baselines, indicating that incorporating serialized graph structures provides additional collaborative information beyond text-only modeling.

Dataset	Method	Training Time	Testing Time	Total
ML-1M	ColdLLM	8.70h	0.40min	8.70h
	FilterLLM	3.40h	0.47min	3.40h
	Ours	1.48h	0.10min	1.48h
Steam	ColdLLM	32.07h	13.43min	32.30h
	FilterLLM	10.95h	6.30min	11.05h
	Ours	3.43h	0.22min	3.43h

Table 2: Comparison of **efficiency** across **LLM-based baselines** of all datasets in our study. The highest efficiency for each dataset is highlighted in **bold**.

5.3 Efficiency Comparison (RQ2)

To evaluate the efficiency of GraphSEA, we compare it with two LLM-based baselines. As shown in Table 2, GraphSEA demonstrates substantially higher training efficiency than the two baselines. Specifically, GraphSEA trains 5.9 and 9.4 times faster than ColdLLM on the ML-1M and Steam datasets, respectively, and 2.3 and 3.2 times faster than FilterLLM. This efficiency gain is mainly at-

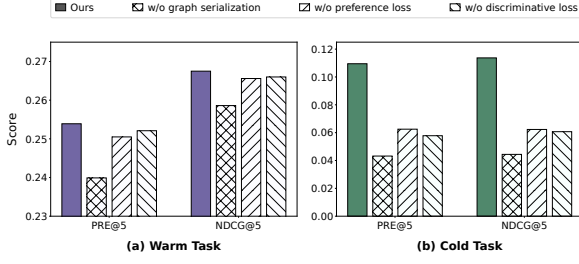


Figure 4: **Ablation study** on key modules of GraphSEA under warm and cold tasks. The x-axis represents PRE@5 and NDCG@5, while the y-axis denotes the corresponding performance scores on ML-1M.

tributed to our graph structure serialization strategy, which treats graph nodes directly as tokens and avoids the complex joint modeling of textual and interaction information.

GraphSEA also achieves higher inference efficiency. During inference, GraphSEA predicts user-item scores through dot-product computations between embeddings. This lightweight calculation avoids the item-by-item inference process required by ColdLLM and the complex probability distribution estimation used by FilterLLM, thereby substantially reducing inference time. This advantage is important for large-scale recommendation, where repeated LLM inference over a large candidate set can incur substantial computational overhead.

5.4 Ablation Study (RQ3)

We conduct an ablation study to examine the contribution of each component in the GraphSEA framework, as shown in Figure 4.

The results show that Graph Structure Serialization plays a crucial role in the model. Its removal leads to a significant drop in performance on the cold item task from 0.1095 to 0.0432. This substantial degradation suggests that textual content alone is insufficient to recover collaborative preference signals when direct interactions are unavailable. By encoding higher-order user-item relations into structured sequences, Graph Structure Serialization provides transferable structural cues for representing cold items. Consequently, it improves the representation quality for cold item feature alignment and substantially enhances the model’s performance on cold item tasks.

In addition to Graph Structure Serialization, the alignment losses also contribute substantially to model performance. Removing the Preference Gap Loss results in a 42.9% performance drop, while removing the Discriminative Consistency Loss leads

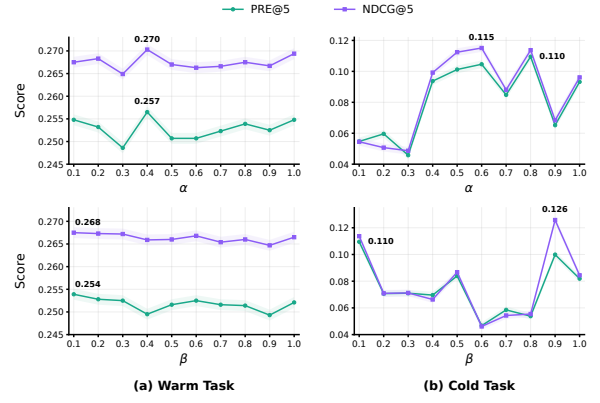


Figure 5: **Parameter analysis** of GraphSEA performance on the ML-1M dataset. PRE@5 and NDCG@5 under varying α (top row) and β (bottom row) for (a) the warm task and (b) the cold task.

to a 47.2% decline. This indicates that the Preference Gap Loss preserves the relative order of items, whereas the Discriminative Consistency Loss plays a crucial role in maintaining the discriminative structure within the embedding space. These two objectives therefore provide complementary supervision by preserving both relative preference ordering and inter-item separability during alignment.

5.5 Hyper-parameter Sensitivity (RQ4)

The hyperparameters in GraphSEA significantly impact the final model performance. We conducted sensitivity experiments on two representative hyperparameters: the weight α for Preference Gap Loss and the weight β for Discriminative Consistency Loss, with results shown in Figure 5.

Overall, GraphSEA achieves a balanced performance across warm and cold tasks with a relatively large α and a small β . A larger α strengthens the model’s ability to preserve preference ordering and generally improves ranking quality, while excessively large values may slightly affect representation learning. In contrast, a small β helps balance semantic consistency and discriminative learning, whereas increasing β tends to cause larger performance fluctuations, particularly on the cold task.

Notably, the warm task remains relatively stable across different values of both hyperparameters, whereas the cold task exhibits substantially larger variations. This indicates that cold-item representations are more sensitive to the weighting of the alignment objectives, as they rely more heavily on collaborative supervision transferred from warm items. In particular, the stronger variation with β suggests that excessive discriminative constraints

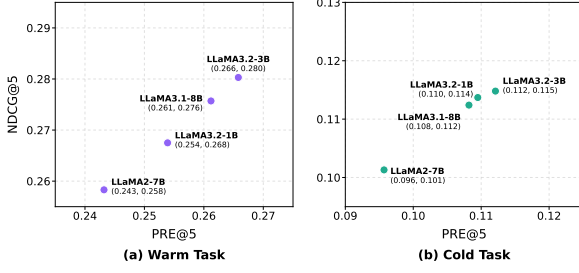


Figure 6: **Comparison of different LLM backbones.** LLaMA variants with different generations and parameter scales are evaluated as the backbone of GraphSEA on both warm and cold item tasks, with performance measured by PRE@5 and NDCG@5.

may perturb the alignment of cold-item representations. These observations highlight the importance of appropriately balancing the two alignment objectives to achieve stable performance.

5.6 LLM Backbone Comparison (RQ5)

To thoroughly evaluate the impact of different architectures, we use LLaMA3.2-1B as the default LLM backbone and conduct controlled comparisons across multiple variants, including LLaMA2-7B, LLaMA3.2-3B, and LLaMA3.1-8B, to assess the generalization performance of GraphSEA. These backbones span different LLaMA generations and parameter scales, ranging from compact 1B and 3B models to larger 7B and 8B variants. All models are evaluated under identical prompt templates to isolate architecture effects. The experimental results are presented in Figure 6.

The experimental results demonstrate that LLaMA3.2-1B offers the best balance between performance and efficiency. Although LLaMA3.2-3B achieves the highest scores in both warm and cold item tasks, LLaMA3.2-1B maintains comparable performance with differences consistently below 5% across all metrics while requiring significantly fewer computational resources. When comparing with LLaMA3.1-8B, the 1B model shows only marginally lower performance on warm items but delivers superior results on cold items. More importantly, it accomplishes this while using just one-eighth of the computational resources required by the 8B model. This combination of strong performance across both task types and excellent operational efficiency makes LLaMA3.2-1B the most practical choice for real-world applications, especially in environments where resource constraints and response times are critical factors.

6 Conclusion

LLM-based cold item recommendation faces two main challenges: (i) difficulty in modeling high-order interactions and preserving the topological structure of interaction graphs, and (ii) the difference between content embeddings derived from textual descriptions and behavior embeddings trained on user interactions, which hinders effective alignment. To address these issues, we propose the GraphSEA framework, which introduces: (i) a graph structure serialization that converts interaction graphs into a structured corpus, enabling LLMs to capture richer collaborative information, and (ii) a cold embedding alignment module with tailored objectives to reduce the discrepancy between content and behavior embeddings, facilitating effective embedding learning. Comprehensive experiments validate the effectiveness and efficiency of GraphSEA in both warm-item and cold-item scenarios, offering a new perspective on using language modeling to encode graph-structured collaborative signals for cold-start recommendation.

Limitations

While GraphSEA shows significant improvements in cold-start recommendation, it still relies on item content. Its performance might significantly degrade for items with sparse, noisy, or low-quality content, as limited content information may weaken the model’s ability to construct reliable item representations. Additionally, the current approach does not fully address the complete cold-start scenario where both users and items are entirely new to the system, which remains a more challenging setting due to the absence of historical interaction signals on both sides.

Ethics Statement

We confirm that we have strictly adhered to the ACL Ethics Policy throughout this study. Our research employs publicly available recommendation datasets, in which user-item interactions and textual descriptions are anonymized, and no personally identifiable information (PII) or private user data is collected or exposed. We do not attempt to identify or re-identify any individual user, and all available information is used solely for academic research purposes. The study does not involve any direct interaction with human participants or real-world deployment. All experiments were conducted entirely in a controlled research environment.

References

- Haoyue Bai, Min Hou, Le Wu, Yonghui Yang, Kun Zhang, Richang Hong, and Meng Wang. 2023. Gorec: a generative cold-start recommendation framework. In *International Conference on Multimedia (MM)*.
- Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Conference on Recommender Systems (RecSys)*.
- Thijmen Bijl, Niels van Weeren, and Suzan Verberne. 2024. Efficient course recommendations with t5-based ranking and summarization. *arXiv preprint arXiv:2406.19018*.
- Desheng Cai, Shengsheng Qian, Quan Fang, Jun Hu, and Changsheng Xu. 2023. User cold-start recommendation via inductive heterogeneous graph neural network. *ACM Transactions on Information Systems (TOIS)*.
- Dong-Kyu Chae, Jin-Soo Kang, Sang-Wook Kim, and Jaeho Choi. 2019. Rating augmentation with generative adversarial networks towards accurate collaborative filtering. In *International World Wide Web Conference (WWW)*.
- Li-Yang Chang, Yuan Fang, Ming-Feng Tsai, and Chuan-Ju Wang. 2026. G-trac: Graph-textual representations alignment for cold-start recommendations. In *International Conference on Web Search and Data Mining (WSDM)*.
- Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology (TIST)*.
- Hao Chen, Yuanchen Bei, Qijie Shen, Yue Xu, Sheng Zhou, Wenbing Huang, Feiran Huang, Senzhang Wang, and Xiao Huang. 2024. Macro graph neural networks for online billion-scale recommender systems. In *International World Wide Web Conference (WWW)*.
- Hao Chen, Zefan Wang, Feiran Huang, Xiao Huang, Yue Xu, Yishi Lin, Peng He, and Zhoujun Li. 2022. Generative adversarial framework for cold-start item recommendation. In *Conference on Information Retrieval (SIGIR)*.
- Kunfeng Chen, Luyao Zhuang, Fei Liao, Juhua Liu, Jian Wang, and Bo Du. 2026. Tool retrieval bridge: Aligning vague instructions with retriever preferences via bridge model. *Neural Networks*.
- Xiaoyu Du, Xiang Wang, Xiangnan He, Zechao Li, Jinhui Tang, and Tat-Seng Chua. 2020. How to learn item representation for cold-start multimedia recommendation? In *International Conference on Multimedia (MM)*.
- Yuyuan Feng, Zhishang Xiang, Chaobin Yang, Qichao Ma, Zerui Chen, Yujing Zhang, Ke Huang, Chuanjie Wu, Zhaoxu Liu, Yili Wang, Xin He, Jiapu Wang, Zijin Hong, Hao Chen, Yuanchen Bei, Kun Wang, Shengyuan Chen, Ningyu Zhang, Enyan Dai, and 16 others. 2026. Graph engineering in the era of llm agents: From individual intelligence to system intelligence. *arXiv preprint arXiv:2608.21156*.
- Wenjing Fu, Zhaohui Peng, Senzhang Wang, Yang Xu, and Jin Li. 2019. Deeply fusing reviews and contents for cold start users in cross-domain recommendation systems. In *Conference on Artificial Intelligence (AAAI)*.
- Mingqi Gao, Xinyu Hu, Xunjian Yin, Jie Ruan, Xiao Pu, and Xiaojun Wan. 2025. Llm-based nlg evaluation: Current status and challenges. *Computational Linguistics*.
- Jyotirmoy Gope and Sanjay Kumar Jain. 2017. A survey on solving cold start problem in recommender systems. In *International Conference on Computing, Communication and Automation (ICCCA)*.
- F. Maxwell Harper and Joseph A. Konstan. 2015. The movielens datasets: History and context. *ACM Transactions on Intelligent Systems and Technology (TIST)*.
- Zijin Hong, Hao Chen, Zheng Yuan, Qinggang Zhang, Luyao Zhuang, Qing Liao, Feiran Huang, Yangqiu Song, and Xiao Huang. 2026a. Errorllm: Modeling sql errors for text-to-sql refinement. In *Conference on Knowledge Discovery and Data Mining (KDD)*.
- Zijin Hong, Hao Wu, Su Dong, Junnan Dong, Yilin Xiao, Yujing Zhang, Zhu Wang, Feiran Huang, Linyi Li, Hongxia Yang, and Xiao Huang. 2026b. Benchmarking llms' mathematical reasoning with unseen random variables questions. In *Conference on Artificial Intelligence (AAAI)*.
- Zijin Hong, Zheng Yuan, Qinggang Zhang, Hao Chen, Junnan Dong, Feiran Huang, and Xiao Huang. 2025. Next-generation database interfaces: A survey of llm-based text-to-sql. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*.
- Feiran Huang, Yuanchen Bei, Zhenghang Yang, Junyi Jiang, Hao Chen, Qijie Shen, Senzhang Wang, Fakhri Karray, and Philip S Yu. 2025. Large language model simulator for cold-start recommendation. In *International Conference on Web Search and Data Mining (WSDM)*.
- Feiran Huang, Zefan Wang, Xiao Huang, Yufeng Qian, Zhetao Li, and Hao Chen. 2023. Aligning distillation for cold-start item recommendation. In *Conference on Information Retrieval (SIGIR)*.

- Manisha Jangid and Rakesh Kumar. 2025. Deep learning approaches to address cold start and long tail challenges in recommendation systems: a systematic review. *Multimedia Tools and Applications*.
- Qing-Yuan Jiang, Yi He, Gen Li, Jian Lin, Lei Li, and Wu-Jun Li. 2019. Svd: A large-scale short video dataset for near-duplicate video retrieval. In *Computer Vision and Pattern Recognition Conference (CVPR)*.
- S Joshua Johnson, M Ramakrishna Murty, and I Navakanth. 2024. A detailed review on word embedding techniques with emphasis on word2vec. *Multimedia Tools and Applications*.
- Hourun Li, Yifan Wang, Zhiping Xiao, Jia Yang, Changling Zhou, Ming Zhang, and Wei Ju. 2025. Disco: graph-based disentangled contrastive learning for cold-start cross-domain recommendation. In *Conference on Artificial Intelligence (AAAI)*.
- Jiaxing Li and Youyong Kong. 2025. Embedding uncertainty modeling for cold-start item recommendation. *Neurocomputing*.
- Lei Li, Yongfeng Zhang, Dugang Liu, and Li Chen. 2023. Large language models for generative recommendation: A survey and visionary discussions. *arXiv preprint arXiv:2309.01157*.
- Hank Liao, Erik McDermott, and Andrew Senior. 2013. Large scale deep neural network acoustic modeling with semi-supervised training data for youtube video transcription. In *IEEE Workshop on Automatic Speech Recognition and Understanding*.
- Jiayan Lin, Yujia Liu, Zijin Hong, Zheng Yuan, Yilin Xiao, Hao Chen, Qinggang Zhang, Xiao Huang, and Feiran Huang. 2026. Are these modules worth their cost? a paradigm-level accuracy-cost analysis of in-context learning text-to-sql. *arXiv preprint arXiv:2608.28432*.
- Xinyu Lin, Wenjie Wang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2024. Bridging items and language: A transition paradigm for large language model-based recommendation. In *Conference on Knowledge Discovery and Data Mining (KDD)*.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*.
- Ruochen Liu, Hao Chen, Yuanchen Bei, Lijia Chen, Qijie Shen, Feiran Huang, Fakhri Karray, and Senzhang Wang. 2026. Setllm: Set large language model for cold-start item recommendation. In *Conference on Knowledge Discovery and Data Mining (KDD)*.
- Ruochen Liu, Hao Chen, Yuanchen Bei, Qijie Shen, Fangwei Zhong, Senzhang Wang, and Jianxin Wang. 2024. Fine tuning out-of-vocabulary item recommendation with user sequence imagination. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Ruochen Liu, Hao Chen, Yuanchen Bei, Zheyu Zhou, Lijia Chen, Qijie Shen, Feiran Huang, Fakhri Karray, and Senzhang Wang. 2025. Filterllm: Text-to-distribution llm for billion-scale cold-start recommendation. *arXiv preprint arXiv:2502.16924*.
- Taichi Liu, Chen Gao, Zhenyu Wang, Dong Li, Jianye Hao, Depeng Jin, and Yong Li. 2023b. Uncertainty-aware consistency learning for cold-start item recommendation. In *Conference on Information Retrieval (SIGIR)*.
- Yunze Luo, Yuezihan Jiang, Yinjie Jiang, Gaode Chen, Jingchi Wang, Kaigui Bian, Peiyi Li, and Qi Zhang. 2025. Online item cold-start recommendation with popularity-aware meta-learning. In *Conference on Knowledge Discovery and Data Mining (KDD)*.
- Feiyang Pan, Shuokai Li, Xiang Ao, Pingzhong Tang, and Qing He. 2019. Warm up cold-start advertisements: Improving ctr predictions via learning to learn id embeddings. In *Conference on Information Retrieval (SIGIR)*.
- Deepak Kumar Panda and Sanjog Ray. 2022. Approaches and algorithms to mitigate cold start problems in recommender systems: a systematic literature review. *Journal of Intelligent Information Systems*.
- Wisam A Qader, Musa M Ameen, and Bilal I Ahmed. 2019. An overview of bag of words; importance, implementation, applications, and challenges. In *International Engineering Conference (IEC)*.
- Ziyu Qiu, Zhilei Zhou, Bradley Niblett, Andrew Johnston, Jeffrey Schwartzentruber, Nur Zincir-Heywood, and Malcolm I Heywood. 2024. Assessing the impact of bag-of-words versus word-to-vector embedding methods and dimension reduction on anomaly detection from log files. *International Journal of Network Management*.
- Dunlei Rong, Lina Yao, Yinting Zheng, Shuang Yu, Xiaofei Xu, Mingyi Liu, and Zhongjie Wang. 2024. Llm enhanced representation for cold start service recommendation. In *International Conference on Service-Oriented Computing*.
- Keigo Sakurai, Ren Togo, Takahiro Ogawa, and Miki Haseyama. 2025. Llm is knowledge graph reasoner: Llm’s intuition-aware knowledge graph reasoning for cold-start sequential recommendation. In *European Conference on Information Retrieval*.
- Rachna Sethi and Monica Mehrotra. 2021. Cold start in recommender systems—a survey from domain perspective. In *Intelligent Data Communication Technologies and Internet of Things (ICICI)*.
- Shaoyun Shi, Min Zhang, Xinxing Yu, Yongfeng Zhang, Bin Hao, Yiqun Liu, and Shaoping Ma. 2019. Adaptive feature sampling for recommendation with missing content feature values. In *International Conference on Information and Knowledge Management (CIKM)*.

- R. Sifa, Anders Drachen, and C. Bauckhage. 2015. Large-scale cross-game player behavior analysis on steam. In *Artificial Intelligence and Interactive Digital Entertainment International Conference (AIIDE)*.
- Changfeng Sun, Han Liu, Meng Liu, Zhaochun Ren, Tian Gan, and Liqiang Nie. 2020. Lara: Attribute-to-feature adversarial learning for new-item recommendation. In *International Conference on Web Search and Data Mining (WSDM)*.
- Albin Uruqi, Iosif Viktoratos, and Athanasios Tsadiras. 2025. A novel framework leveraging large language models to enhance cold-start advertising systems. *Future Internet*.
- Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30.
- Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. 2017. Dropoutnet: Addressing cold start in recommender systems. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Jianling Wang, Haokai Lu, James Caverlee, Ed H Chi, and Minmin Chen. 2024a. Large language models as data augmenters for cold-start item recommendation. In *International World Wide Web Conference (WWW)*.
- Wenbo Wang, Ben Chen, Bingquan Liu, Xinxin Wang, Luwei Yang, Wen Jiang, Wei Ning, and Jian Guan. 2024b. Mutual information assisted graph convolution network for cold-start recommendation. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
- Wenbo Wang, Bingquan Liu, Lili Shan, Chengjie Sun, Ben Chen, and Jian Guan. 2024c. Preference aware dual contrastive learning for item cold-start recommendation. In *Conference on Artificial Intelligence (AAAI)*.
- Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2024d. Learnable item tokenization for generative recommendation. In *International Conference on Information and Knowledge Management (CIKM)*.
- Xinghua Wang, Zhaohui Peng, Senzhang Wang, Philip S Yu, Wenjing Fu, Xiaokang Xu, and Xiaoguang Hong. 2020. Cdlfm: cross-domain recommendation for cold-start users via latent feature mapping. *Knowledge and Information Systems*.
- Jian Wei, Jianhua He, Kai Chen, Yi Zhou, and Zuoyin Tang. 2017. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications*.
- Yinwei Wei, Xiang Wang, Qi Li, Liqiang Nie, Yan Li, Xuanping Li, and Tat-Seng Chua. 2021. Contrastive learning for cold-start recommendation. In *International Conference on Multimedia (MM)*.
- Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, Hui Xiong, and Enhong Chen. 2024. A survey on large language models for recommendation. *World Wide Web*.
- Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. 2020. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*.
- Yunjia Xi, Weiwen Liu, Jianghao Lin, Xiaoling Cai, Hong Zhu, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, and Yong Yu. 2024. Towards open-world recommendation with knowledge augmentation from large language models. In *Conference on Recommender Systems (RecSys)*.
- Yilin Xiao, Jin Chen, Qinggang Zhang, Yujing Zhang, Chuang Zhou, Longhao Yang, Lingfei Ren, Xin Yang, and Xiao Huang. 2026a. Logicpoison: Logical attacks on graph retrieval-augmented generation. In *Association for Computational Linguistics (ACL)*.
- Yilin Xiao, Junnan Dong, Chuang Zhou, Su Dong, Qianwen Zhang, Di Yin, Xing Sun, and Xiao Huang. 2025. Graphrag-bench: Challenging domain-specific reasoning for evaluating graph retrieval-augmented generation. *arXiv preprint arXiv:2506.02404*.
- Yilin Xiao, Chuang Zhou, Qinggang Zhang, Bo Li, Qing Li, and Xiao Huang. 2026b. Reliable reasoning path: Distilling effective guidance for LLM reasoning with knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*.
- Yilin Xiao, Chuang Zhou, Yujing Zhang, Qinggang Zhang, Su Dong, Shengyuan Chen, Chang Yang, and Xiao Huang. 2026c. Lag: Logic-augmented generation from a cartesian perspective. *arXiv preprint arXiv:2508.05509*.
- Yilin Xiao, Zhehan Zhu, Yujing Zhang, Jin Chen, Zijin Hong, Luyao Zhuang, Qinggang Zhang, Shengyuan Chen, Xiaocao Ouyang, Lingfei Ren, and Xiao Huang. 2026d. Zero-mem: Zero-token memory operations for llm agents. *arXiv preprint arXiv:2607.29377*.
- Xiaoxiao Xu, Chen Yang, Qian Yu, Zhiwei Fang, Jiaxing Wang, Chaosheng Fan, Yang He, Changping Peng, Zhangang Lin, and Jingping Shao. 2022. Alleviating cold-start problem in ctr prediction with a variational embedding learning framework. In *International World Wide Web Conference (WWW)*.

- Chang Yang, Xinrun Wang, Junzhe Jiang, Qinggang Zhang, and Xiao Huang. 2026a. LLM-based world models can make decisions solely, but rigorous evaluations are needed. *Transactions on Machine Learning Research (TMLR)*.
- Chang Yang, Xinrun Wang, Qinggang Zhang, Qi Jiang, and Xiao Huang. 2025. Efficient integration of external knowledge to llm-based world models via retrieval-augmented generation and reinforcement learning. In *Findings of Empirical Methods in Natural Language Processing (EMNLP)*.
- Chang Yang, Chuang Zhou, Yilin Xiao, Su Dong, Luyao Zhuang, Yujing Zhang, Zhu Wang, Zijin Hong, Zheng Yuan, Zhishang Xiang, Shengyuan Chen, Huachi Zhou, Qinggang Zhang, Ninghao Liu, Jinsong Su, Xinrun Wang, Yi Chang, and Xiao Huang. 2026b. Graph-based agent memory: Taxonomy, techniques, and applications. *arXiv preprint arXiv:2602.05665*.
- Haotong Yang, Xiyuan Wang, Qian Tao, Shuxian Hu, Zhouchen Lin, and Muhan Zhang. 2024. Gf-fusion: Rethinking the combination of graph neural network and large language model. *arXiv preprint arXiv:2412.06849*.
- Hongyu Yao, Zijin Hong, Hao Chen, Zhiqing Li, Qijie Shen, Zuobin Ying, Qihua Feng, Huan Gong, and Feiran Huang. 2026. Macro graph of experts for billion-scale multi-task recommendation. In *Conference on Knowledge Discovery and Data Mining (KDD)*.
- Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Conference on Knowledge Discovery and Data Mining (KDD)*.
- An Zhang, Yang Deng, Yankai Lin, Xu Chen, Ji-Rong Wen, and Tat-Seng Chua. 2024. Large language model powered agents for information retrieval. In *Conference on Information Retrieval (SIGIR)*.
- Qinggang Zhang, Shengyuan Chen, Yuanchen Bei, Zheng Yuan, Huachi Zhou, Zijin Hong, Hao Chen, Yilin Xiao, Chuang Zhou, Junnan Dong, Yi Chang, and Xiao Huang. 2025a. A survey of graph retrieval-augmented generation for customized large language models. *arXiv preprint arXiv:2501.13958*.
- Weizhi Zhang, Yuanchen Bei, Liangwei Yang, Henry Peng Zou, Peilin Zhou, Aiwei Liu, Yinghui Li, Hao Chen, Jianling Wang, Yu Wang, Feiran Huang, Sheng Zhou, Jiajun Bu, Allen Lin, James Caverlee, Fakhri Karray, Irwin King, and Philip S. Yu. 2025b. Cold-start recommendation towards the era of large language models (llms): A comprehensive survey and roadmap. *arXiv preprint arXiv:2501.01945*.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Zican Dong, Yupeng Hou, Beichen Zhang, Yingqian Min, Junjie Zhang, Peiyu Liu, Xiaolei Wang, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, and 4 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.
- Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, and Qing Li. 2024. Recommender systems in the era of large language models (llms). *IEEE Transactions on Knowledge and Data Engineering (TKDE)*.
- Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. 2024. Adapting large language models by integrating collaborative semantics for recommendation. In *International Conference on Data Engineering (ICDE)*.
- Huachi Zhou, Jiahe Du, Yujing Zhang, Luyao Zhuang, Chang Yang, Zijin Hong, Jiaqi Bai, Qinggang Zhang, Kaixiong Zhou, and Xiao Huang. 2026. Relu: Refined chunk embeddings learning for ultra-long snp genomic prediction in crop breeding. In *Conference on Knowledge Discovery and Data Mining (KDD)*.
- Hai Zhu, Yantao Yu, Lei Shen, Bing Wang, and Xiaoyi Zeng. 2026. Gatesid: Adaptive gating for semantic-collaborative alignment in cold-start recommendation. *arXiv preprint arXiv:2603.22916*.
- Yongchun Zhu, Kaikai Ge, Fuzhen Zhuang, Ruobing Xie, Dongbo Xi, Xu Zhang, Leyu Lin, and Qing He. 2021a. Transfer-meta framework for cross-domain recommendation to cold-start users. In *Conference on Information Retrieval (SIGIR)*.
- Yongchun Zhu, Ruobing Xie, Fuzhen Zhuang, Kaikai Ge, Ying Sun, Xu Zhang, Leyu Lin, and Juan Cao. 2021b. Learning to warm up cold item embeddings for cold-start recommendation with meta scaling and shifting networks. In *Conference on Information Retrieval (SIGIR)*.
- Ziwei Zhu, Jingu Kim, Trung Nguyen, Aish Fenton, and James Caverlee. 2021c. Fairness among new items in cold start recommender systems. In *Conference on Information Retrieval (SIGIR)*.
- Ziwei Zhu, Shahin Sefati, Parsa Saadatpanah, and James Caverlee. 2020. Recommendation for new users and new items via randomized training and mixture-of-experts transformation. In *Conference on Information Retrieval (SIGIR)*.
- Luyao Zhuang, Shengyuan Chen, Yilin Xiao, Huachi Zhou, Yujing Zhang, Hao Chen, Qinggang Zhang, and Xiao Huang. 2026a. Linearrag: Linear graph retrieval augmented generation on large-scale corpora. In *International Conference on Learning Representations (ICLR)*.
- Luyao Zhuang, Qinggang Zhang, Huachi Zhou, Yujing Zhang, and Xiao Huang. 2026b. Losemb: Logic-guided semantic bridging for inductive tool retrieval. In *International World Wide Web Conference (WWW)*.

A Appendix

A.1 Algorithm and Theoretical Analysis

This section provides supplementary algorithmic and theoretical details of GraphSEA. Specifically, we first present the complete pseudocode of the proposed framework and then provide a theoretical analysis to justify the preservation of graph structural information during graph serialization.

A.1.1 Pseudocode of GraphSEA

The overall pipeline of the proposed **GraphSEA** framework consists of two main modules: (i) LLM-Based Embedding Generation, which systematically transforms the user-item interaction graph into a structured corpus to generate behavior embeddings, while leveraging pre-trained LLMs to produce high-quality content embeddings from textual descriptions; and (ii) Cold Embedding Alignment, which effectively aligns behavior and content embeddings across multiple facets, including interaction probability, preference order, and item distinction.

Specifically, the two modules are executed as follows. Given the user-item interaction graph, GraphSEA first performs random walks from each node to construct a serialized sequence corpus, which is then used to optimize the behavior LoRA adapter on the shared frozen LLM backbone and derive behavior embeddings for users and warm items. In parallel, a separate content LoRA adapter is optimized on the textual description corpus to obtain content embeddings for users and items. During cold embedding alignment, the behavior embeddings of warm items are partially masked and concatenated with their content embeddings to simulate the absence of behavioral information for cold items. The alignment module is optimized by jointly considering recommendation accuracy, preference-order consistency, and item distinction. At inference time, the unavailable behavior embedding of each cold item is fully masked, and its content embedding is used with the learned alignment functions to predict user-item preference scores. The complete procedure is summarized in Algorithm 1.

A.1.2 Theorems and Proof

Theorem 1 ensures that graph structure faithfully preserves the high-order behavioral relationships present in the original user-item interaction graph.

Algorithm 1 The Overall Workflow of GraphSEA.

Input: User-item interaction graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{D})$; User set $\mathcal{U}$; Cold-item set $\mathcal{I}_c$; Warm-item set $\mathcal{I}_w$; Textual descriptions $\mathcal{D}$.

Parameter: Frozen LLM backbone θ_{LLM} ; Behavior and content LoRA parameters $\theta_{\text{LoRA}}^{(B)}$ and $\theta_{\text{LoRA}}^{(C)}$; α, β, r .

Output: User and item representations for cold-item recommendation.

```

1: # LLM-Based Embedding Generation
2:  $\mathcal{S} \leftarrow \emptyset$  //Initialize sequence corpus
3: for  $v \in \mathcal{V}$  do
4:    $s_v \leftarrow \text{RandomWalk}(v)$  //Eq. (2)
5:    $\mathcal{S} \leftarrow \mathcal{S} \cup s_v$ 
6: end for
7:  $\theta_{\text{LoRA}}^{(B)} \leftarrow \theta_{\text{LoRA}}^{(B)} - \lambda \nabla_{\theta_{\text{LoRA}}^{(B)}} \mathcal{L}_{\text{pre}}(\mathcal{S})$  //Eq. (3)
8:  $E_U^{(B)}, E_{I_w}^{(B)} \leftarrow f_{\theta_{\text{LLM}}, \theta_{\text{LoRA}}^{(B)}}(\mathcal{S})$  //Eqs. (4)–(6)
9:  $E_U^{(C)}, E_I^{(C)} \leftarrow f_{\theta_{\text{LLM}}, \theta_{\text{LoRA}}^{(C)}}(\mathcal{D})$  //Eqs. (7)–(8)
10: # Cold Embedding Alignment
11: while not converged do
12:   # Concatenation and Masking
13:    $E_U \leftarrow E_U^{(B)} \oplus E_U^{(C)}$  //Eq. (9)
14:    $E_{I_w}^{(B)} \leftarrow M \odot E_{I_w}^{(B)} // M \sim \text{Bernoulli}(1-r), \text{Eq. (10)}$ 
15:    $E_{I_w} \leftarrow E_{I_w}^{(B)} \oplus E_{I_w}^{(C)}$  //Eq. (11)
16:   # Compute alignment losses
17:    $\mathcal{L}_{\text{rec}} \leftarrow \{r_t, r_p\}$  //Eq. (12)
18:    $\mathcal{L}_{\text{gap}} \leftarrow \{\sigma(r_{t,+} - r_{t,-}), \sigma(r_{p,+} - r_{p,-})\}$  //Eq. (13)
19:    $\mathcal{L}_{\text{disc}} \leftarrow \{d_t, d_p\}$  //Eqs. (14)–(15)
20:   # Total loss
21:    $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{rec}} + \alpha \cdot \mathcal{L}_{\text{gap}} + \beta \cdot \mathcal{L}_{\text{disc}}$  //Eq. (16)
22:   Update parameters by minimizing  $\mathcal{L}_{\text{total}}$ 
23: end while
24: for  $i \in \mathcal{I}_c$  do
25:    $E_i \leftarrow \mathbf{0} \oplus E_i^{(C)}$  //Fully mask the missing behavior embedding
26:   Predict user-item preference scores with the learned alignment functions.
27: end for

```

Theorem 1. Let $P_s(v_{i+1}|v_i; n)$ denote the empirical conditional probability computed from n sampled random walks, and let $P_g(v_{i+1}|v_i)$ represent the true transition probability in the original graph. As the number of sampled random walks approaches infinity, we have:

$$\lim_{n \rightarrow \infty} P_s(v_{i+1}|v_i; n) = P_g(v_{i+1}|v_i). \quad (17)$$

This theorem ensures that graph structure faithfully preserves the high-order behavioral relationships present in the original user-item interaction graph.

Proof. The empirical conditional probability $P_s(v_j|v_i)$ is derived from random walks and represents the probability of the co-occurrence of node v_i followed by node v_j . It is defined as:

$$P_s(v_j | v_i) = \frac{C_s(v_i \rightarrow v_j)}{\sum_{k \in N(v_i)} C_s(v_i \rightarrow v_k)}, \quad (18)$$

where $C_s(v_i \rightarrow v_j)$ counts the co-occurrences of v_i followed by v_j in sampled walk sequences, and $N(v_i)$ represents the set of neighboring nodes of v_i considered during random-walk transitions.

For the co-occurrences count $C_s(v_i \rightarrow v_j)$ from v_i to v_j , its expected value is:

$$\mathbb{E}[C_s(v_i \rightarrow v_j)] = d_i \cdot l \cdot P_g(v_j | v_i), \quad (19)$$

where d_i is the node degree, l denotes the walk length, and $P_g(v_j | v_i)$ denotes the transition probability in the graph.

According to the Law of Large Numbers, when either the number of samples $n \rightarrow \infty$ or $d_i \rightarrow \infty$, the co-occurrence count $C_s(v_i \rightarrow v_j)$ converges to its expected value:

$$\frac{C_s(v_i \rightarrow v_j)}{d_i \cdot l} \xrightarrow{\text{a.s.}} P_g(v_j | v_i). \quad (20)$$

The total number of transitions originating from node v_i is $\sum_{k \in N(v_i)} C_s(v_i \rightarrow v_k)$, and its expected value is:

$$\begin{aligned} \mathbb{E}\left[\sum_{k \in N(v_i)} C_s(v_i \rightarrow v_k)\right] &= \sum_{k \in N(v_i)} \mathbb{E}[C_s(v_i \rightarrow v_k)] \\ &= d_i \cdot l. \end{aligned} \quad (21)$$

By substituting the expected values into the definition of the empirical conditional probability, where both the numerator and denominator provably converge to their respective true expectations, we obtain:

$$\begin{aligned} P_s(v_j | v_i) &\xrightarrow{\text{a.s.}} \frac{\mathbb{E}[C_s(v_i \rightarrow v_j)]}{\mathbb{E}[\sum_{k \in N(v_i)} C_s(v_i \rightarrow v_k)]} \\ &= P_g(v_j | v_i). \end{aligned} \quad (22)$$

□

The theorem guarantees the mathematical soundness of the graph structure serialization process. When sufficient sampling is performed, the node co-occurrence frequencies within the sequence corpus precisely capture the topological characteristics of the original graph structure. As a result, the behavior embeddings acquired by the LLM through next-token prediction effectively encode authentic higher-order behavioral patterns. This fundamental property accounts for GraphSEA's superior performance compared to conventional approaches limited to first-order interaction modeling in cold-start recommendation scenarios.

A.2 Experimental Details

A.2.1 Dataset Process

We evaluate GraphSEA on both warm and cold items using the ML-1M and Steam datasets.

- **MovieLens-1M** is a movie dataset released by GroupLens Research that contains approximately 1 million ratings on a 5-star scale from 6,040 users on 3,706 movies, along with user information and movie metadata.
- **Steam** is a gaming interaction dataset sourced from the Steam platform that captures user behaviors including purchases, playtime, and reviews, while providing rich game metadata such as genres and developers.

For each dataset, following the setup in prior research, 20% of the items are designated as cold items, and their interaction data is split into a cold-start validation set and a test set in a 1:1 ratio. The interaction records of the remaining 80% regular items are divided into a training set, validation set, and test set in an 8:1:1 ratio.

A.2.2 Baseline Details

To evaluate the effectiveness of our GraphSEA, we conducted a comparative analysis of 10 state-of-the-art models in the field of cold-start recommendation. The comparison was carried out on two different datasets. The models we compared can be categorized into the following three groups: (i) Dropout-based models: DropoutNet (Volkovs et al., 2017), MTPR (Du et al., 2020), Heater (Zhu et al., 2020), and CLCRec (Wei et al., 2021). (ii) Generative-based models: DeepMusic (Van den Oord et al., 2013), MetaEmb (Pan et al., 2019), GAR (Chen et al., 2022), and ALDI (Huang et al., 2023). (iii) LLM-based models: ColdLLM (Huang et al., 2025) and FilterLLM (Liu et al., 2025).

- **DropoutNet** uses input dropout during training to combine content features and latent representations for robust recommendations.
- **MTPR** employs dual item representations and a multi-task pairwise ranking loss to model relationships between warm and cold items.
- **Heater** maps auxiliary information to collaborative representation space using combined training, randomized training, and mixture-of-experts transformation.

- **CLCRec** maximizes mutual information between collaborative embeddings and content features through contrastive learning.
- **DeepMusic** utilizes deep convolutional neural networks to predict latent factors from audio signals for music recommendation.
- **MetaEmb** uses meta-learning to generate initial ID embeddings for new ads by simulating cold start and warm up phases with gradient-based optimization.
- **GAR** employs adversarial training between the generator and the recommender to generate cold item embeddings with a distribution similar to warm embeddings.
- **ALDI** leverages aligning distillation to transfer collaborative information from warm items to cold items by modeling and reducing their differences in rating distribution, ranking, and identification.
- **ColdLLM** uses a coupled funnel framework to filter users, enabling LLMs to simulate interactions efficiently for cold-start recommendations.
- **FilterLLM** adopts a "Text-to-Distribution" paradigm to predict user interaction distributions for cold items in a single inference.

A.2.3 Implementation Details

We adopt the official implementations of each baseline model for reproduction. For all baseline methods not based on large language models, the embedding dimension is consistently set to 200. The extensive experiments are conducted on an NVIDIA Tesla V100-SXM2 GPU with 32GB of memory. For a fair comparison, we measure the running time on a single GPU, distinguishing between training and testing phases. All experiments are repeated five times, and the reported results are averaged over the five runs.

The maximum number of training epochs for our GraphSEA model is set to 20, with the learning rate fixed at 0.001. The hyperparameters are dynamically adjusted based on the dataset: for the ML-1M dataset, the values of α and β are set to 0.8 and 0.1, respectively, while for the Steam dataset, the values of α and β are set to 0.4 and 0.1, respectively.

For embedding generation, we use the same pre-trained LLM backbone with two independently optimized LoRA adapters for behavior and content

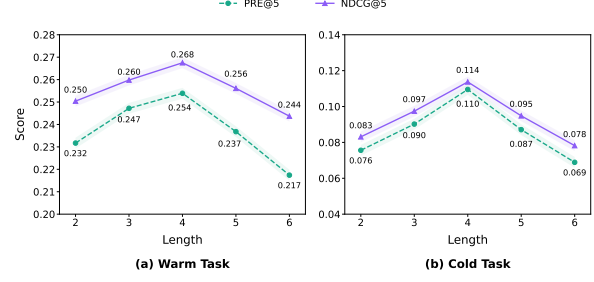


Figure 7: **Analysis on random walk length.** The x-axis denotes the length ranging from 2 to 6, where length 4 corresponds to a random walk path with four nodes.

representation learning. Specifically, the behavior adapter is trained on the serialized user-item interaction sequences to generate behavior embeddings, while the content adapter is trained on the natural-language metadata corpus to generate content embeddings. The underlying LLM backbone is shared and kept frozen during LoRA fine-tuning. Accordingly, references to LLM fine-tuning in the main text denote parameter-efficient LoRA fine-tuning rather than full-parameter fine-tuning of the backbone.

For prompts, we design dataset-specific templates to structure the available information in a natural language format. These templates are crafted to maximize the utility of available metadata while maintaining a conversational tone. For MovieLens-1M, we leverage rich user demographic information such as age, gender, occupation, location and movie characteristics like title, genres, while for Steam, we incorporate comprehensive game attributes including pricing, publisher information, and technical specifications:

- **ML-1M:** “This is user {User_ID}, who is a {Age}-year-old {Gender} working as a {Occupation} and living in the {Zip-code} area. The item with ID {Item_ID} is titled {Title} and belongs to the following genres: {Genres}.”
- **Steam:** “This is user {User_ID}. The Steam game with ID {Item_ID} is titled {Title} and belongs to the {Genres}. It is published by {Publisher} and priced at {Price}. The game specifications include: {Specs}.”

A.3 Supplementary Results

Beyond the ablation studies of three core modules and the sensitivity analysis of two loss weights, we further investigate three additional critical factors

Strategy	ML-1M Warm		ML-1M Cold	
	PRE@5	NDCG@5	PRE@5	NDCG@5
BFS	0.251	0.265	0.092	0.095
DFS	0.252	0.266	0.105	0.112
Ours	0.254	0.268	0.110	0.114

Table 3: Comparison of **different sampling strategies** on both warm and cold item tasks.

that influence model performance. Additionally, in Appendix A.3.5, we investigate the embedding differences before and after alignment.

A.3.1 Random Walk Length Analysis

The random walk length l is a predefined sampling hyperparameter in graph structure serialization, which denotes the number of node tokens sampled in each random-walk sequence. We examined random walk lengths $l \in \{2, 3, 4, 5, 6\}$, with experimental results presented in Figure 7.

The results show that random walk length significantly impacts recommendation performance. In both warm-item and cold-item tasks, performance exhibits an initial increase followed by a decline as the random walk length increases, reaching optimal performance at length 4. This indicates that sequences that are too short cannot adequately capture user-item interaction information, while excessively long sequences may introduce noise and dilute key information, leading to suboptimal results. Based on these empirical findings, we set the random walk length as 4 in our framework to achieve optimal recommendation performance.

This indicates that overly short walks provide limited structural context, making it difficult to capture higher-order user-item collaborative patterns. In contrast, overly long walks may include weakly related nodes and introduce noisy collaborative signals, thereby diluting the useful local structural information. Based on these empirical findings, we set the random-walk length to 4 in GraphSEA.

A.3.2 Sampling Strategy Analysis

The choice of graph sampling strategy significantly impacts the quality of the generated sequence corpus and, consequently, the model’s ability to capture collaborative information. To thoroughly evaluate different sampling approaches, we compare our adopted random walk strategy against two classical graph traversal methods: Breadth-First Search (BFS) and Depth-First Search (DFS). The experimental results are presented in Table 3.

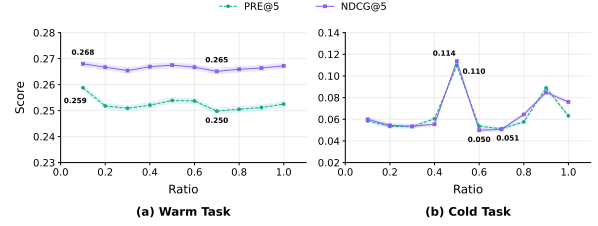


Figure 8: **Analysis on Masking Ratio.** The x-axis shows masking ratio r , which is the Bernoulli distribution parameter for token masking, where $r=0.5$ means each token has a 50% chance of being masked.

Random walk consistently outperforms both BFS and DFS across all metrics. The key reason lies in how each method samples paths from the graph. BFS and DFS follow fixed rules where BFS always explores neighbors layer by layer, while DFS always goes as deep as possible before backtracking. These rigid strategies mean that certain types of paths are sampled much more frequently than others, regardless of their actual importance in the graph. In contrast, random walk samples paths based on the graph’s natural structure where frequently connected nodes appear more often in the corpus, while rare connections appear less often. This produces a training corpus where path frequencies match the actual interaction patterns, allowing the LLM to learn embeddings that accurately reflect real user-item relationships.

A.3.3 Masking Ratio Sensitivity

The input masking ratio r controls the robustness of feature learning. As shown in Figure 8, we systematically vary r from 0.1 to 1.0 to analyze its effect on recommendation performance.

For warm items, both PRE@5 and NDCG@5 metrics remain relatively stable across different masking ratios, fluctuating within the narrow ranges of 0.26 to 0.27 and 0.25 to 0.253, respectively. This stability suggests that the masking strategy has minimal impact on items with abundant historical interaction data, as the model can learn robust representations from rich behavioral patterns.

For cold items, the performance exhibits a distinct inverted U-shaped curve, with both PRE@5 and NDCG@5 peaking at 0.5. Performance degrades significantly when the masking ratio is either too low or too high. This is because insufficient masking fails to encourage effective content feature learning, while excessive masking removes too much critical information and compromises

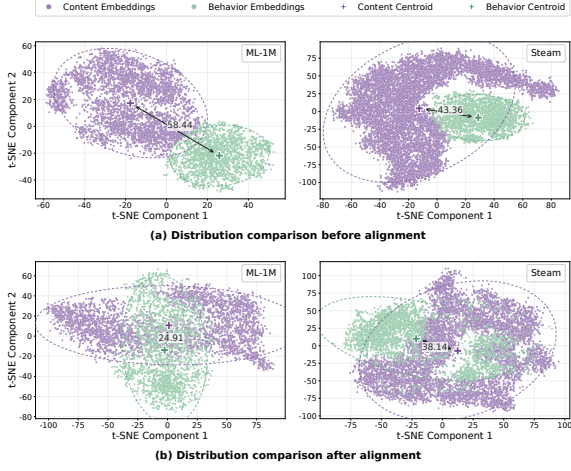


Figure 9: Visualization of **Content Embeddings** and **Behavior Embeddings** on Two Datasets. (a) shows the distinct distributions before alignment, while (b) demonstrates increased overlap between embeddings after alignment.

representation quality.

A.3.4 Embedding alignment differences

Figure 9 (a), which corresponds to Figure 2 in the main text, illustrates the visualization of content embeddings and behavior embeddings on two datasets: ML-1M and Steam. The embeddings are reduced to two dimensions using t-SNE to analyze their distributional patterns. The clear separation between the two embedding types highlights their distinct nature.

To further validate the effectiveness of our alignment module, we visualize the embedding distributions after applying our three-faceted alignment objectives. Figure 9 (b) shows the t-SNE visualization of content and behavior embeddings after alignment. Notably, the two embedding types exhibit significantly more overlap compared to their distributions before alignment, demonstrating the effectiveness of our alignment strategy.

This improved alignment can be attributed to our multi-faceted optimization approach. The Score Reconstruction Loss $\mathcal{L}_{\text{rec}}$ directly minimizes the MSE between target scores derived from behavior embeddings and predicted scores from concatenated embeddings, effectively forcing the model to learn compatible representations. The Preference Gap Loss $\mathcal{L}_{\text{gap}}$ ensures that the relative ordering between positive and negative samples is preserved across both embedding spaces, maintaining the collaborative filtering principles essential for recommendation quality. Most importantly, the Discrimi-

native Consistency Loss $\mathcal{L}_{\text{disc}}$ preserves the semantic relationships and discriminative structure within the embedding space, ensuring that while the distributions become more aligned, the meaningful distinctions between different items are retained.

The increased overlap indicates that our alignment module successfully bridges the gap between content and behavior embeddings, enabling effective knowledge transfer from warm items with rich interactions to cold items. This distributional alignment is crucial for cold-start recommendation, as it enables the model to generate effective representations for cold items even in the absence of interactions, thereby supporting accurate and reliable recommendations.

A.4 Frequently Asked Questions (FAQs)

A.4.1 How does the graph structure serialization preserve collaborative information?

Theorem 1 proves that random walk sampling on the user-item interaction graph in GraphSEA converges to true transition probabilities as sample size increases, with a detailed proof provided in Appendix A.1.2. This theoretical guarantee ensures the converted token sequences retain high-order collaborative information while maintaining computational tractability. The LLM then learns these patterns via next-token prediction, effectively capturing both direct user-item interactions and higher-order neighborhood structures. Crucially, this approach overcomes the limitation of traditional prompt-based methods that only consider first-order interactions, enabling more comprehensive representation learning for cold items.

A.4.2 Does the sampling time of GraphSEA on the raw data count as part of the training time?

In the GraphSEA framework, the time for random walk sampling is indeed included in the training time, but the sampling process itself is not time-consuming. This is due to the inherent sparsity of the user-item interaction graph in recommendation scenarios, combined with GraphSEA’s design of short-path random walks with a fixed length of 4. As a result, the sampling process only requires a constant number of lightweight neighbor-selection operations, which can be further optimized through parallelization. Thus, even on large-scale datasets, the time spent on the sampling phase is negligible compared to the subsequent LLM training process

and does not become a bottleneck for system efficiency. Therefore, the sampling time is included in the overall training time in the paper without being separately reported.

A.4.3 Why was random walk chosen as the sequence sampling method for user-item interaction graphs, and were other sampling methods attempted?

Random walks offer theoretically guaranteed convergence and effectively capture higher-order collaborative relationships in user-item interaction graphs. With linear time complexity, random walks preserve the local stochastic characteristics of user behavior sequences while capturing crucial higher-order interactions through multi-step transitions. These properties make them particularly suitable for sequence prediction tasks in the GraphSEA framework, which is based on LLMs.

In addition, we conducted comparative experiments with other graph traversal strategies, including DFS and BFS, and the detailed results are reported in Appendix A.3.2. The experimental results show that, compared with DFS and BFS, random walks better preserve the probabilistic characteristics of the original graph while generating more diverse sequences. This makes them more suitable for meeting the sequence modeling requirements of LLMs, leading to superior overall performance.

A.4.4 How are cold items handled differently during training and inference?

During training, the behavior embeddings of warm items are randomly masked to simulate the absence of collaborative information in cold-start scenarios. Specifically, a subset of warm-item behavior embeddings is replaced by zero vectors and concatenated with their corresponding content embeddings. This allows the alignment module to learn how to reconstruct effective recommendation representations when behavioral information is unavailable.

During inference, cold items do not have behavior embeddings because no historical interactions are available. Therefore, their missing behavior component is fully masked. The learned item mapping function then transforms this representation into the aligned recommendation space, where user-item preference scores can be computed together with the learned user representations. In this way, the masking mechanism used during training explicitly simulates the representation form encountered by cold items at inference time.

A.4.5 Why are separate LoRA adapters used for behavior and content embedding generation?

Behavior and content embeddings originate from substantially different information sources. Behavior embeddings are learned from serialized user-item interaction sequences and are expected to capture collaborative and structural patterns, whereas content embeddings are derived from natural-language descriptions and primarily encode semantic information. Directly optimizing a single adapter for both objectives may couple these heterogeneous learning signals and reduce the specialization of the resulting representations.

Therefore, GraphSEA employs two independently optimized LoRA adapters while sharing the same frozen LLM backbone. The behavior adapter is trained on serialized graph sequences, whereas the content adapter is trained on textual metadata. This design preserves a common underlying representation space provided by the shared backbone while allowing each adapter to specialize in its corresponding information modality. The resulting representations are subsequently integrated and aligned by the Embedding Alignment module.

A.5 Future Research Directions

The significant performance improvements on both movie and game datasets demonstrate our method’s adaptation to different recommendation scenarios. These successes open up promising research directions, such as extending the framework to handle more complex scenarios like complete cold-start cases where both users and items are new. Beyond recommendations, our graph structure serialization shows great potential for broader applications in various graph-based tasks, where complex graph structures can be effectively transformed into structured node-token sequences for LLMs to process.