Structuring and Tokenizing Distributed User Interest Context for Generative Recommendation

Ruizhong Qiu*
University of Illinois
Urbana–Champaign, IL, USA
rq5@illinois.edu

Yinglong Xia
Meta MRS
Menlo Park, CA, USA
yxia@meta.com

Dongqi Fu
Meta MRS
Menlo Park, CA, USA
dongqifu@meta.com

Hanqing Zeng
Meta MRS
Menlo Park, CA, USA
zengh@meta.com

Ren Chen
Meta MRS
Menlo Park, CA, USA
renchen@meta.com

Xiangjun Fan
Meta MRS
Menlo Park, CA, USA
maxfan@meta.com

Hong Li
Meta MRS
Menlo Park, CA, USA
hongli@meta.com

Hong Yan
Meta MRS
Menlo Park, CA, USA
hong@meta.com

Hanghang Tong
University of Illinois
Urbana–Champaign, IL, USA
htong@illinois.edu

Abstract

Generative recommendation is an emerging paradigm that has shown promise in industrial recommendation systems, aiming to predict users' next interactions from their historical behaviors. At the core of generative recommendation lies item tokenization, which bridges item semantics and recommendation models. However, existing methods often struggle to effectively *organize* and *inject* complex user-behavioral and item-semantic contexts into recommendation models simultaneously. On the one hand, existing graph-based integration methods, such as graph serialization and graph neural networks, either suffer from scalability issues or exploit only local graph information. On the other hand, existing semantic tokenization methods typically rely on heuristics and lack explicit supervision signals, which may lead to inaccurate or suboptimal semantic representations. To address these limitations in user interest context modeling, we propose **G2Rec**, a scalable framework that unifies holistic graph-based user co-engagement modeling with semantic tokenization for industrial-scale generative recommendation. **First**, we construct a sparsified *item-item co-engagement graph* with $O(M \log M)$ edges as the item schema, where M denotes the total number of interactions. **Second**, we design a scalable "soft" clustering algorithm with time complexity $O(\rho M \log M)$ per iteration to extract distributed *interest prototypes* from the constructed graph, where ρ is a small constant representing the sparsity of the soft cluster membership distribution rather than a hard one-to-one assignment. **Third**, based on the *interest prototypes* and *item interest profiles* extracted by soft clustering, we tokenize these profiles together with users' interested items to train a generative sequential recommendation model. Overall, G2Rec enables recommendation models to capture holistic and semantically grounded user interest prototypes without requiring ground-truth user interests, thereby providing more comprehensive and accurate modeling of user behavior contexts in industrial sequential recommendation. Online deployment across product surfaces and extensive experiments on public datasets demonstrate the superiority of G2Rec over existing methods. Proofs of theoretical results can be found at https://arxiv.org/abs/2606.20554.

CCS Concepts

• **Information systems** → **Recommender systems**; • **Mathematics of computing** → **Graph algorithms**.

Keywords

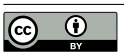
Generative Recommendation, Tokenization, Graph Clustering

ACM Reference Format:
Ruizhong Qiu, Yinglong Xia, Dongqi Fu, Hanqing Zeng, Ren Chen, Xiangjun Fan, Hong Li, Hong Yan, and Hanghang Tong. 2026. Structuring and Tokenizing Distributed User Interest Context for Generative Recommendation. In *20th ACM Conference on Recommender Systems (RecSys '26), September 27-October 02, 2026, Minneapolis, MN, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3773078.3831928

1 Introduction

Generative recommendation (GR) [7, 9, 10, 15, 70] has emerged as a promising paradigm in the field of industrial recommender systems. By leveraging the autoregressive nature of user behavior, GR aims to predict the next interactions of users based on their historical contexts using large language model (LLM) architectures, thereby providing users with a more personalized and responsive experience. This approach has shown considerable potential in various scenarios, including e-commerce, online advertising, and content streaming, where understanding the sequential nature of user behavior is crucial for delivering relevant recommendations. By modeling the temporal dependencies between user interactions, GR seeks to capture the underlying preferences and interests that drive

*Work done during an internship at Meta.



RecSys '26, Minneapolis, MN, USA

ACM ISBN 979-8-4007-2284-4/2026/09
https://doi.org/10.1145/3773078.3831928

user behavior, ultimately leading to more accurate and effective recommendations.

At the core of GR is item tokenization to bridge item semantics and the GR model. However, existing GR methods often struggle to effectively incorporate complex user-behavioral and item-semantic contexts simultaneously into the recommendation model. On the one hand, while there exist graph integration methods to leverage the relational information from user-item relational graphs, they suffer from scalability issues or only utilize local graph information. For instance, graph serialization methods transform the graph into a sequence, but the long serialization can incur expensive computation of LLMs. Graph neural networks (GNNs), on the other hand, typically perceive only a small subgraph for each user, but they cannot fully exploit the holistic graph information. These limitations hinder the effectiveness of graph-based methods in industry-scale generative sequential recommendation. Another parallel line of work is *semantic tokenization*, which aims to represent each item as a few semantic tokens. However, they typically rely on heuristic learning objectives and lack supervision signals for semantic tokens. Consequently, these methods may not be guaranteed to learn accurate semantic representations, leading to suboptimal performance in recommendation tasks.

To address these critical limitations in this complex context modeling, we propose *Sparse Co-Engagement Graph Schema for Generative Recommendation* (**G2Rec**), a novel method that bridges holistic graph-based co-engagement modeling and semantic tokenization in industry-scale generative sequential recommendation. Systematically, our method G2Rec first constructs an item-item co-engagement graph, capturing the complex relationships between user behavior and items with empirical efficiency and theoretical accuracy. We then design a "soft" clustering algorithm in G2Rec to obtain item interest prototypes from the constructed graph, where each cluster represents an interest prototype, and each item serving as a node has the soft cluster membership distribution over clusters (but not a hard-assigned one-to-one exclusive membership). Finally, based on the item interest profiles extracted from the soft clustering, G2Rec tokenizes the item semantics along with the original user engagement sequence to train the generative sequential recommendation models.

Our proposed method G2Rec offers several advantages over existing methods. Firstly, it provides a more comprehensive and accurate modeling of user behavior by incorporating both graph-based co-engagement contexts and semantic item representations. Secondly, our method is highly scalable, allowing for efficient processing of large-scale graphs and user sequences. Finally, the use of clustering algorithms ensures that the learned semantic tokens are accurate and meaningful, without relying on explicit labels or heuristics. We conduct extensive experiments on public datasets and online deployment on product surfaces to evaluate the performance of G2Rec. The results demonstrate the superiority of our method over many existing generative recommendation methods, highlighting its potential for real-world applications in industry-scale generative recommendation.

Main contributions. We summarize the main contributions of this work as follows.

- **Sparse graph schema as co-engagement contexts.** We propose a sparsified item co-engagement graph as the item schema. We show that sampling $O(M \log M)$ co-engagements suffices to approximately preserve structural information (Theorem 2), where M is the total number of interactions. This is substantially sparser than the original co-engagement graph of quadratic size $O(M^2)$.
- **Scalable soft graph clustering.** We propose a scalable soft graph clustering algorithm. It is based on our proposed differentiable modularity objective function, which can be computed in nearly linear time $O(\rho M \log M)$ and can be accelerated by GPUs, where ρ is a small sparsity constant.
- **Clustering-based item interest profiling.** We propose (i) using the clusters of the item-item co-engagement graph as item interest prototypes and (ii) leveraging the soft cluster membership of each item as the item interest profile for user behavioral information beyond item feature similarity.
- **Interest profile tokenization for GR.** We introduce a novel representation of user interaction history sequences: alternating between items and item interest profiles, allowing the generative recommender to effectively learn user interest transition patterns.
- **Strong online performance.** Large-scale online A/B tests on Meta's product surfaces show the superiority of G2Rec.

2 Preliminaries

2.1 Notations

Let $\mathcal{U}$ denote the complete set of users, where each user is represented as $u \in \mathcal{U}$. Similarly, let $\mathcal{I}$ denote the universe of items, with $i \in \mathcal{I}$ denoting a particular item. Each item $i \in \mathcal{I}$ has an embedding vector $x_i \in \mathbb{R}^d$, where d denotes the embedding dimensionality. Let $X := [x_1, \ldots, x_{|\mathcal{I}|}]^\top \in \mathbb{R}^{|\mathcal{I}| \times d}$ denote the item embedding matrix. For each user $u \in \mathcal{U}$, we record the past interactions of the user u as a sequence $\mathcal{I}_u = [i_1, \ldots, i_N]$ of items, where N denotes the number of items that user u has interacted with in the past, and each $i_j \in \mathcal{I}$ $(j = 1, \ldots, N)$ denotes an item that user u has interacted with. We call sequence $\mathcal{I}_u$ the *interaction history* of user u and assume $\bigcup_{u \in \mathcal{U}} \mathcal{I}_u = \mathcal{I}$ to avoid cold start. Let $M := \sum_{u \in \mathcal{U}} |\mathcal{I}_u|$ denote the total number of interactions. These interactions can be of various forms (such as clicks, views, comments, and purchases), and they reflect the user behavioral patterns to be learned by the recommender system.

2.2 Problem Definition

The objective of a generative recommender system is to predict an item that a user will probably interact with next, from a sequence of items that the user has interacted with in the past. The task is formally defined in Problem 1.

Problem 1 (generative sequential recommendation). ***Input:*** *(i) a user $u \in \mathcal{U}$; (ii) an integer N specifying the length of the interaction history of user u; (iii) a sequence $\mathcal{I}_u = [i_1, \ldots, i_N]$ of items that user u has interacted with in the past; (iv) the embedding x_i of each item $i \in \mathcal{I}_u$.* ***Output:*** *rank the candidate items $\mathcal{I}$ by the likelihood of each item to be the next interaction of user u.*

3 Proposed Framework: G2Rec

In this section, we introduce our proposed method *Sparse Co-Engagement Graph Schema for Generative Recommendation* (G2Rec). We describe our graph construction procedure in Section 3.1, our new scalable soft graph clustering method in Section 3.2, and our graph-based co-engagement tokenization in Section 3.3. We will discuss industry-scale deployment in Section 4.

3.1 User Behavior Modeling via a Sparsified Item Co-Engagement Graph

In this subsection, we describe how we construct a graph to capture user behavioral patterns. We will use the graph for item interest profiling in Section 3.2.

In existing graph-based recommendation methods, user behavior is often modeled using the user-item bipartite graph. In the bipartite graph, users and items are represented as nodes, and edges connect users to the items they have interacted with. However, for industry-scale applications with massive user bases, this can be extremely computationally expensive, especially when the user has a long interaction history.

Co-engagement graph. To address this critical limitation, we propose eliminating user nodes from the constructed graph and model user behavior using an item-item graph instead. By eliminating users from the graph, we can focus on the more static item relationships, reducing the need for frequent updates. In this work, we propose the item-item *co-engagement* graph as a natural definition of the item-item graph. The two items $i, j \in \mathcal{I}$ are said to be *co-engaged* if there exists a user $u \in \mathcal{U}$ such that u has interacted with both items i and j (i.e., $i \in \mathcal{I}_u$ and $j \in \mathcal{I}_u$); the user u here is not necessarily unique. The co-engagement graph captures the patterns of items being interacted with together, allowing us to model user behavior without explicitly representing users.

Formally, in the item-item co-engagement graph $\mathcal{G}^* = (\mathcal{I}, \mathcal{E}^*)$, items $\mathcal{I}$ serve as nodes of $\mathcal{G}^*$, and co-engagements $\mathcal{E}^*$ serve as edges of $\mathcal{G}^*$. Let $\mathcal{E}^* \subseteq \mathcal{I} \times \mathcal{I}$ denote the set of item-item co-engagements:

$$\mathcal{E}^* := \bigcup_{u \in \mathcal{U}} \mathcal{I}_u \times \mathcal{I}_u = \{(i, j) : \exists u \in \mathcal{U} \text{ s.t. } i \in \mathcal{I}_u,\ j \in \mathcal{I}_u\}, \quad (1)$$

where $\times$ denotes the Cartesian product between sets. The item-item co-engagement graph $\mathcal{G}^*$ provides a powerful tool for modeling user behavior patterns beyond item feature similarity. For example, consider two items: a smartphone and a phone case. While they may not be similar in terms of their features, they are often co-engaged by users who purchase a new phone and then look for accessories to go with it. The co-engagement graph can capture such patterns, providing insightful information that complements item feature information.

From a theoretical perspective, the interaction history sequence of each user can be represented as a path on the item-item co-engagement graph, as formally stated in Observation 1. Hence, edges on the item-item co-engagement graph encode the user interest transition behaviors.

Observation 1 (expressiveness). *For any user $u \in \mathcal{U}$, their interaction history sequence $\mathcal{I}_u = [i_1, \ldots, i_N]$ is a path on the co-engagement graph $\mathcal{G}^* = (\mathcal{I}, \mathcal{E}^*)$. That is, $(i_t, i_{t+1}) \in \mathcal{E}^*$ for all $t = 1, \ldots, N-1$.* □

Graph sparsification. Nonetheless, since $|\mathcal{E}^*|$ can be as large as $O(M^2)$, using the original graph $\mathcal{G}^*$ would be computationally prohibitive in industrial scenarios. To ensure scalability, we propose a theoretically grounded approach to sparsifying the graph. Our goal is to approximately preserve the graph Laplacian, which encodes essential structural information of the graph [36]. Formally, given $m \ll N^2$, we define the sparsified co-engagement graph as $\mathcal{G} := (\mathcal{I}, \mathcal{E})$ where we sample only m co-engagements to include in $\mathcal{E}$ for each user:

$$\mathcal{E} := \bigcup_{u \in \mathcal{U}} \text{sample}(\mathcal{I}_u \times \mathcal{I}_u,\ m), \quad (2)$$

where $\text{sample}(\cdot, m)$ denotes sampling to size m with replacement. We will use the sparsified graph $\mathcal{G}$ for item interest profiling in Section 3.2.

As implied by the following Theorem 2, we theoretically show that $|\mathcal{E}| = O(M \log M)$ suffices to preserve the graph Laplacian, which is nearly linear w.r.t. the total number M of interactions. Notably, this is substantially sparser than $|\mathcal{E}^*| = O(M^2)$.

Theorem 2 (nearly linear complexity). *Suppose that every user has N interactions. Let $L_{\mathcal{E}^*}$ be the Laplacian of $\mathcal{E}^*$, and let $L_{\mathcal{E}}$ be the Laplacian of $\mathcal{E}$ with edge weights $\frac{N(N-1)}{2m}$ to match the total edge weights of L^*. Given any $0 < \delta < 1$ and $\epsilon > 0$, if we use*

$$m := \left\lceil 2N\left(\frac{1}{3\epsilon} + \frac{1}{\epsilon^2}\right) \log \frac{2|\mathcal{I}|}{\delta} \right\rceil,$$

then $|\mathcal{E}| \le O(M \log M)$, and with probability at least $1 - \delta$,

$$(1 - \epsilon) L_{\mathcal{E}^*} \preceq L_{\mathcal{E}} \preceq (1 + \epsilon) L_{\mathcal{E}^*},$$

where $\preceq$ denotes the Loewner order.

3.2 Item Interest Profiling via Soft Graph Clustering

In this subsection, we elaborate on how we leverage the item-item co-engagement graph to compute item interest profiles.

A key challenge in processing the co-engagement graph $\mathcal{G}$ is that some co-engagement relationships can be random. To address this challenge, we propose to employ graph clustering to extract meaningful co-engagements. The intuition is that if a group of items is frequently co-engaged with each other (i.e., they belong to the same cluster in the graph), their co-engagements are more likely to be meaningful. Hence, we propose to use the clusters of the co-engagement graph $\mathcal{G}$, which should provide reliable information for the recommendation model to learn. We call these clusters **item interest prototypes**, which can be interpreted as the underlying interest categories. Furthermore, as shown in Observation 1, the co-engagement graph encodes user interest transition behaviors. Therefore, by treating each cluster as an item interest prototype, we can help the recommendation model to gain insights into the underlying interest transitions that drive user behavior.

However, since existing graph clustering algorithms (e.g., Louvain [3], Leiden [50]) typically assume each node belongs to only one cluster, they are unsuitable for capturing complex interests where each item can correspond to multiple interest prototypes simultaneously. For instance, a short-form video of an avocado toast recipe may attract both users interested in "healthy eating" and those interested in "morning routines". To bridge this gap, we

propose to employ **soft graph clustering** instead, which allows for continuous cluster memberships and differentiable optimization.

Nevertheless, existing soft graph clustering methods (e.g., [66]) are typically not sufficiently scalable for industrial use cases. To enable scalable soft graph clustering, we propose a differentiable objective for soft graph clustering that generalizes the classic notion of graph modularity [34], which allows end-to-end gradient-based optimization and can thus be solved efficiently on GPUs.

For each item $i \in \mathcal{I}$, we aim to find a membership distribution $p_i \in \mathbb{R}^C$, where C denotes the number of interest prototypes, and $p_{i,a}$ represents the probability that item i belongs to interest prototype $a \in \{1, \ldots, C\}$. We call p_i the **item interest profile** of item i. To ensure $\sum_{a=1}^{C} p_{i,a} = 1$, we use parameterization $p_i := \mathrm{softmax}(z_i)$, where logits $z_i \in \mathbb{R}^C$ are the variables to be optimized. We initialize and sparsify the variables using the Leiden algorithm [50]. Collectively, $p_1, \ldots, p_{|\mathcal{I}|}$ form a *soft membership* matrix $P := [p_1, \ldots, p_{|\mathcal{I}|}]^\top \in \mathbb{R}^{|\mathcal{I}| \times C}$.

Next, we introduce our differentiable objective for soft graph clustering. Let $A \in \{0,1\}^{|\mathcal{I}| \times |\mathcal{I}|}$ denote the adjacency matrix of the co-engagement graph $\mathcal{G}$, and let $k \in \mathbb{N}^{|\mathcal{I}|}$ denote the degree vector (i.e., $k_i := \sum_j A_{i,j}$ is the degree of each item $i \in \mathcal{I}$). Recall that given a hard (cluster) membership $h \in \{1, \ldots, C\}^{|\mathcal{I}|}$ (i.e., h_i is the interest prototype that item i belongs to), the classic graph modularity Q_{hard} [34] is defined as

$$Q_{\text{hard}}(h) := \frac{1}{|\mathcal{E}|} \sum_{i,j \in \mathcal{I}} \left(A_{i,j} - \gamma \frac{k_i k_j}{|\mathcal{E}|} \right) 1_{[h_i = h_j]}, \tag{3}$$

where $\frac{k_i k_j}{|\mathcal{E}|}$ is the expected number of edges between i, j in a random graph under the Newman–Girvan null model theory [35], and $\gamma > 0$ denotes the clustering resolution. Here, we adopt the duplicate representation of the undirected graph $\mathcal{G}$, i.e., (i, j) and (j, i) are both in $\mathcal{E}$.

However, we cannot directly use Q_{hard} to optimize a soft membership P because $1_{[h_i = h_j]}$ is not a differentiable operation. To address the non-differentiability, we propose to maximize the expected modularity under distribution P as a differentiable objective, which we show admits a simple closed form.

Proposition 3 (closed form). *The expected modularity is*

$$Q_{\text{soft}}(P) := \mathop{\mathbb{E}}_{h \sim P}[Q_{\text{hard}}(h)] = \left(\frac{1}{|\mathcal{E}|} \sum_{(i,j) \in \mathcal{E}} p_i^\top p_j \right) - \gamma \frac{\|P^\top k\|_2^2}{|\mathcal{E}|^2}. \tag{4}$$

We call this objective Q_{soft} the *soft modularity*. It can be computed efficiently on GPUs when $\mathcal{E}$ and P are sparse, as shown in Proposition 4.

Proposition 4 (nearly linear complexity). *If each row of P has $\le \rho$ nonzero entries, then $Q_{\text{soft}}(P)$ can be computed in time*

$$\mathrm{O}(\rho |\mathcal{E}|) = \mathrm{O}(\rho M \log M). \tag{5}$$

The soft modularity objective Q_{soft} can be interpreted as follows. The inner product $p_i^\top p_j = \sum_{a=1}^{C} p_{i,a} p_{j,a}$ represents the probability that items i and j belong to the same interest prototype under the soft membership distribution P, serving as a continuous analog to the term $1_{[h_i = h_j]}$ in hard modularity. The intuition behind Q_{soft} is to encourage high co-membership probabilities $p_i^\top p_j$ for item pairs (i, j) that are heavily co-engaged, while penalizing over-membership to item pairs that are not expected to be connected according to the Newman–Girvan null model theory [35]. Furthermore, unlike the original modularity Q_{hard}, our proposed Q_{soft} is fully differentiable w.r.t. P, making it suitable as an objective function for optimizing the item interest profiles P.

3.3 Interest Profile Tokenization for Generative Sequential Recommendation

In this subsection, we describe how to incorporate item interest profiles into generative sequential recommendation.

Interest profile tokenization. As soft item interest profiles are continuous, we use continuous tokens to represent interest profiles. Before introducing our interest profile tokenization, we first introduce the tokenization of item interest prototypes. For each interest prototype $a \in \{1, \ldots, C\}$, we define the interest prototype embedding $v_a \in \mathbb{R}^d$ of a as the weighted average of the embeddings of the items of interest prototype a:

$$v_a := \frac{\sum_{i \in \mathcal{I}} p_{i,a} x_i}{\sum_{i \in \mathcal{I}} p_{i,a}}, \qquad a \in \{1, \ldots, C\}. \tag{6}$$

Let $V := [v_1, \ldots, v_C]^\top \in \mathbb{R}^{C \times d}$ denote the interest prototype embedding matrix. Then, Equation (6) can be rewritten into the matrix form as $V = \frac{P^\top X}{P^\top 1}$ and can be computed efficiently as P is sparse.

Next, we define the *interest profile token* y_i of each item $i \in \mathcal{I}$ as the weighted average of embeddings v_a of prototypes a of item i:

$$y_i := \sum_{a=1}^{C} p_{i,a} v_a, \qquad i \in \mathcal{I}. \tag{7}$$

Let $Y := [y_1, \ldots, y_{|\mathcal{I}|}]^\top \in \mathbb{R}^{|\mathcal{I}| \times d}$ denote the interest profile token matrix. Then, Equation (7) can be rewritten into the matrix form as $Y = PV$, which can be computed efficiently as P is sparse.

Sequence formulation. Our key idea is to design a new sequence format that takes item interest profiles into account. Specifically, given the interaction history sequence $\mathcal{I}_u = [i_1, \ldots, i_N]$ of a user $u \in \mathcal{U}$, we define a *user interest transition sequence* as follows:

$$\mathcal{R}_u := [\langle \texttt{BOS} \rangle, x_{i_1}, y_{i_1}, \ldots, x_{i_N}, y_{i_N}], \tag{8}$$

where $\langle \texttt{BOS} \rangle$ denotes a beginning sequence. Here, $y_{i_1}, \ldots, y_{i_N}$ can be interpreted as the graph-based semantic tokenization of the items $i_1, \ldots, i_N$, respectively.

Training objectives. Let F denote the generative sequential recommender that autoregressively predicts the next-token distribution. Our goal here is to predict the next item $i_{t+1} \in \mathcal{I}$ that user u will interact with next, given the user interaction history sequence $[i_1, \ldots, i_t]$ $(t = 1, 2, \ldots)$. Hence, for each $t = 1, \ldots, N-1$, we use the cross entropy loss w.r.t. to train the generative recommender F:

$$\mathcal{L}_{\text{item}}^t := -\log F(i_{t+1} \mid \langle \texttt{BOS} \rangle, x_{i_1}, y_{i_1}, \ldots, x_{i_t}, y_{i_t}) \tag{9}$$

$$= -\log F(i_{t+1} \mid (\mathcal{R}_u)_{\le 2t+1}). \tag{10}$$

Furthermore, to help the recommender understand user interest transition behaviors, we also train the recommender to predict the interest of each interaction. Since the ground-truth user interests are not available, we instead propose using the item interest profile

Table 1: Statistics of public datasets.

Dataset	#Users	#Items	#Interacts	Sparsity
Beauty	22,363	12,101	198,502	99.93%
Sports	25,598	18,357	296,337	99.95%
Toys	19,412	11,924	167,597	99.93%
Yelp	30,431	20,033	316,354	99.95%

as the soft label in the cross entropy loss for each $t = 1, \ldots, N$:

$$\mathcal{L}^t_{\text{profile}} := -\sum_{a=1}^{C} p_{i_t,a} \log F(a \mid \langle \text{BOS} \rangle, x_{i_1}, y_{i_1}, \ldots, x_{i_t}) \tag{11}$$

$$= -\sum_{a=1}^{C} p_{i_t,a} \log F(a \mid (\mathcal{R}_u)_{\leq 2t}). \tag{12}$$

Together, we train the recommender F using a weighted combination of the two losses simultaneously:

$$\mathcal{L}^t := \mathcal{L}^t_{\text{item}} + \lambda \mathcal{L}^t_{\text{profile}}, \tag{13}$$

where $\lambda > 0$ is a hyperparameter to control the weight of $\mathcal{L}^t_{\text{profile}}$.

4 Industrial Deployment

We have successfully productionized our G2Rec on multiple product surfaces at Meta. Considering the huge user base (billions of monthly active users [17]) of Meta products (e.g., Instagram Reels), a main challenge in deployment lies in scalability and real-time efficiency. To ensure the timeliness of item interest profiles, we have implemented an in-house high performance graph processing engine that can efficiently execute the graph clustering algorithm [3] in a distributed environment. We run the graph clustering algorithm periodically offline, so the execution time of the graph clustering algorithm has no impact on the response time to users.

5 Experiments

We conduct extensive experiments on both public datasets and our product surfaces to answer the following research questions:

- **RQ1:** How does the proposed G2Rec compare with state-of-the-art recommendation methods offline?
- **RQ2:** How does the proposed G2Rec perform on Meta's product surfaces online?
- **RQ3:** How does the proposed G2Rec influence the efficiency in training and inference, respectively?
- **RQ4:** How do the proposed soft graph clustering and item interest profiling influence recommendation quality?

5.1 Experimental Settings

In this subsection, we describe the experimental settings in our offline empirical evaluation on public datasets.

Public datasets. To evaluate our proposed G2Rec, we conduct experiments on four widely-used public real-world datasets: Beauty, Sports, Toys, and Yelp. Beauty, Sports, and Toys are sub-categories from the Amazon review data collection [33], which is collected from Amazon, a popular online shopping platform. Yelp is another popular online platform that provides crowd-sourced reviews about businesses, and the Yelp dataset [1] is collected from Yelp. In our experiments, we use Yelp data from January 1st, 2019 onwards. The dataset statistics are summarized in Table 1. We can see that all of these datasets are highly sparse, which ensures that these datasets are sufficiently challenging for evaluating the performance under the sequential recommendation setting.

Baseline methods. To comprehensively benchmark the performance of our proposed G2Rec, we compare our G2Rec with six strong baseline methods, including classic methods, sequential / tokenization-based methods, and graph-based methods. We briefly describe the baseline methods as follows. (i) Classic methods: **POP** is a heuristic method that ranks items based on their popularity, which is known to be a strong baseline in recommendation (e.g., [64]). Matrix factorization **MF** [20] is the most classic recommendation method; we use the Bayesian Personalized Ranking (BPR) loss [45] here. (ii) Sequential methods: **GRU4Rec** [14] utilizes the GRU recurrent neural network [6] for modeling sequences and subsequently makes recommendation predictions. **SASRec** [19] uses multi-head self-attention [52] to handle intricate sequential data. **BERT4Rec** [48] uses the cloze objective function from BERT [8] to enable self-supervised learning instead of autoregressively predicting only the next item. **Caser** [49] combines both horizontal and vertical convolution operations to more effectively capture complex interactions within user interaction history sequences. **EAGER** [57] is a two-stream generative recommender with behavior-semantic collaboration. (iii) Graph-based methods. **LightGCN** [12] is a strong graph convolutional network for collaborative filtering. **HeLLM** [11] uses hypergraphs to enhance LLM-based recommenders.

Evaluation protocol & metrics. For sequential recommendation evaluation, we sort user interactions by their timestamps in ascending order to form user interaction history sequences. To ensure a reliable evaluation, we employ a 5-core setting to exclude items with fewer than 5 interactions, following previous works (e.g., [46, 48]). Following previous works (e.g., [19, 44], we adopt the leave-one-out approach to split the dataset. Specifically, for the interaction history sequence of each user, we used the last item as the test data, the second-to-last item as the validation data, and the rest of the sequence as the training data. To evaluate the ranking capability of our proposed G2Rec, we form the test set as follows: for each user, we use the last item they interacted with as the positive item and randomly sample 99 items from the remaining items as negative items. We use Recall@1, Recall@5, Recall@10, NDCG@5, NDCG@10, and the mean reciprocal rank (MRR) as metrics.

Implementation details. We use Llama 2 13B as the recommender model backbone and finetune it for 3 epochs using low-rank adapters (LoRA) with 16 ranks and rank dropout rate 0.05. We initialize the model parameters using the Xavier initialization and optimize the model parameters using the Adam optimizer with initial learning rate 0.0003 and the cosine learning rate schedule with 100 warmup steps. We use SASRec embeddings with $d = 64$ as item embeddings and limit the maximum sequence length to 50 for all methods on all datasets to avoid out-of-memory errors and to ensure a fair comparison. To control the number of interest prototypes, we use $\gamma = 0.8$ for Beauty and Toys and $\gamma = 1$ for Sports and Yelp. For baseline methods, we use the open-source code released by their authors and adapt the code to our experimental settings.

Table 2: Comparison with baseline methods on public datasets (best marked in bold). Our proposed G2Rec consistently outperforms all baseline methods on all four datasets.

Method Type →		Classic		Sequential / Tokenization-Based					Graph-Based		
Dataset ↓	**Metric ↓**	POP	MF	GRU4Rec	SASRec	BERT4Rec	Caser	EAGER	LightGCN	HeLLM	G2Rec (Ours)
Beauty	Recall@1	0.0678	0.0405	0.1870	0.1531	0.1337	0.1337	0.1213	0.1435	0.1690	**0.2067**
	Recall@5	0.2105	0.1461	0.3741	0.3640	0.3032	0.3125	0.2678	0.3081	0.2802	**0.3917**
	Recall@10	0.3386	0.2311	0.4696	0.4739	0.3942	0.4106	0.3663	0.4042	0.3413	**0.4892**
	NDCG@5	0.1391	0.0934	0.2848	0.2622	0.2219	0.2268	0.1962	0.2286	0.2278	**0.3035**
	NDCG@10	0.1803	0.1207	0.3156	0.2975	0.2512	0.2584	0.2278	0.2596	0.2474	**0.3334**
	MRR	0.1558	0.1096	0.2852	0.2614	0.2263	0.2308	0.2060	0.2340	0.2359	**0.3034**
Sports	Recall@1	0.0763	0.0489	0.1455	0.1255	0.1135	0.1160	0.0417	0.1226	0.1021	**0.1750**
	Recall@5	0.2293	0.1603	0.3466	0.3375	0.2866	0.3055	0.1398	0.2841	0.2214	**0.3903**
	Recall@10	0.3423	0.2491	0.4622	0.4722	0.4014	0.4299	0.2232	0.3855	0.2946	**0.5093**
	NDCG@5	0.1538	0.1048	0.2497	0.2341	0.2020	0.2126	0.0906	0.2056	0.1641	**0.2869**
	NDCG@10	0.1902	0.1334	0.2869	0.2775	0.2390	0.2527	0.1174	0.2383	0.1875	**0.3254**
	MRR	0.1660	0.1202	0.2520	0.2378	0.2100	0.2191	0.1078	0.2133	0.1750	**0.2867**
Toys	Recall@1	0.0585	0.0257	0.1878	0.1262	0.1114	0.0997	0.1201	0.1310	0.1507	**0.2006**
	Recall@5	0.1977	0.0978	0.3682	0.3344	0.2614	0.2795	0.2717	0.2799	0.2717	**0.3779**
	Recall@10	0.3008	0.1715	0.4663	0.4493	0.3540	0.3896	0.3777	0.3721	0.3375	**0.4691**
	NDCG@5	0.1286	0.0614	0.2820	0.2327	0.1885	0.1919	0.1977	0.2078	0.2147	**0.2931**
	NDCG@10	0.1618	0.0850	0.3136	0.2698	0.2183	0.2274	0.2319	0.2374	0.2359	**0.3225**
	MRR	0.1430	0.0819	0.2842	0.2338	0.1967	0.1973	0.2075	0.2154	0.2228	**0.2942**
Yelp	Recall@1	0.0801	0.0624	0.2375	0.2405	0.2188	0.2053	0.0976	**0.2696**	0.2148	0.2558
	Recall@5	0.2415	0.2036	0.5745	0.5976	0.5111	0.5437	0.2903	0.5517	0.4053	**0.6105**
	Recall@10	0.3609	0.3153	0.7373	0.7597	0.6661	0.7265	0.4088	0.6756	0.4909	**0.7736**
	NDCG@5	0.1622	0.1333	0.4113	0.4252	0.3696	0.3784	0.1960	0.4165	0.3160	**0.4398**
	NDCG@10	0.2007	0.1692	0.4642	0.4778	0.4198	0.4375	0.2343	0.4566	0.3436	**0.4927**
	MRR	0.1740	0.1470	0.3927	0.4026	0.3595	0.3630	0.2009	0.4025	0.3142	**0.4180**
Average Rank →		8.62	9.75	2.46	2.83	6.35	5.23	7.94	4.50	6.27	**1.04**

5.2 Offline Testing

To answer RQ1, we compare our proposed G2Rec with baseline methods to evaluate the recommendation performance. The results presented in Table 2 and Figure 1 demonstrate the effectiveness of our proposed method G2Rec in comparison to all six baseline methods across all four public datasets. Our G2Rec consistently outperforms all baseline methods across all six metrics on all four datasets, achieving the highest recall, NDCG, and MRR. We observe that the performance gap between our G2Rec and the best-performing baseline method is significant. For instance, our proposed G2Rec achieves 14.9% higher NDCG@5 than the best-performing baseline on the Sports dataset. As these datasets are large-scale datasets with diverse user behavior patterns, the results suggest that our method is capable of handling complex user behavior data and can effectively capture the underlying patterns. In terms of average rank, our G2Rec achieves the top rank across all datasets, indicating its overall superiority over the baseline methods; in stark contrast, the average rank of the baseline methods varies across datasets. Overall, the results demonstrate the effectiveness of our proposed G2Rec in modeling user behavior patterns and item relationships using the co-engagement graph. This suggests that our G2Rec can model item relationships and user behavior patterns using the co-engagement graph and provides a substantial advantage over existing methods.

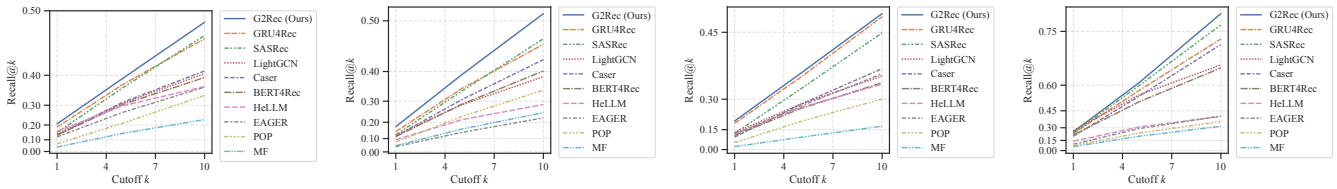
Table 3: Training and inference time per batch. Our interest profile tokens have only negligible impact on training and inference time.

Method	Training	Inference
Item-Only	0.763	0.0534
G2Rec (Ours)	0.806	0.0561
Overhead	+0.043	+0.0027

5.3 Large-Scale Online Testing

We have productized our proposed G2Rec on multiple product surfaces at Meta. While some internal modifications were made to adapt it for deployment on each product surface, the core methodology remains consistent with the proposed framework.

To answer RQ2, we evaluate our proposed G2Rec via both short-term (7-day) and long-term online A/B testing. (Due to the corporate policy, we are unable to disclose the detailed settings.) We have observed **consistent improvement** in terms of a wide range of top-line metrics on **user engagement**, **content diversity**, and **serving efficiency**. In particular, product launches with our holistic interest modeling lead to $> 0.03\%$ in-session improvement and exhibit solid wins (0.06% to 0.19%) in terms of multiple user engagement metrics



(a) Beauty. **(b) Sports.** **(c) Toys.** **(d) Yelp.**

Figure 1: Our G2Rec consistently outperforms all baselines on all four datasets under various cutoffs.

Table 4: Ablation study on soft graph clustering. Our soft graph clustering algorithm consistently outperforms hard graph clustering on all datasets in terms of modularity.

Method	Beauty	Sports	Toys	Yelp
Hard (Leiden)	0.419	0.365	0.437	0.691
Soft (Ours)	**0.499**	**0.452**	**0.550**	**0.757**

such as total time-spent, likes, shares, etc. These results provide strong evidence that our proposed G2Rec considerably enhances user experience by accurately capturing user behavioral patterns and interest prototypes.

5.4 Efficiency Study

In addition to evaluating the effectiveness of our proposed G2Rec, we also investigate its efficiency as efficiency is important in industrial scenarios. Table 3 reports the training and inference time per batch for our method and the item-only baseline on the Beauty dataset. The results show that incorporating interest profile tokens into the model has only a negligible impact on both training and inference time. Specifically, our method requires only 0.043 seconds more per batch during training and only 0.0027 seconds more per batch during inference compared to the item-only baseline. This suggests that our approach can be efficiently integrated into industrial recommendation systems without incurring significant computational overhead. This efficiency is due to the fact that our interest profile tokens are computed offline using the co-engagement graph, and then simply concatenated with the item embeddings during training and inference. This design allows us to leverage the structural information captured by the co-engagement graph without requiring expensive online computations. Overall, these results demonstrate that our approach offers a good balance between effectiveness and efficiency, making it a practical solution for industrial recommender systems.

5.5 Ablation Studies

To answer RQ4, we conduct ablation studies on our proposed soft graph clustering and item interest profiling.

Efficacy of soft graph clustering. We conduct an ablation study to evaluate the efficacy of our soft graph clustering algorithm. We compare our method with hard graph clustering produced by the Leiden algorithm [50]. As shown in Table 4, the results demonstrate that our soft graph clustering algorithm consistently outperforms hard graph clustering in terms of modularity, a widely-used metric for evaluating the quality of graph clustering. The improvements are significant across all datasets (for example, 23.8% improvement in modularity score on the Sports dataset). This suggests that our soft graph clustering algorithm is able to capture more nuanced and complex item interest profiles in the co-engagement graph, which is essential for accurately modeling user interests. These results validate the design choice of using soft graph clustering in our proposed G2Rec and highlight its importance in achieving high-quality item interest profiles.

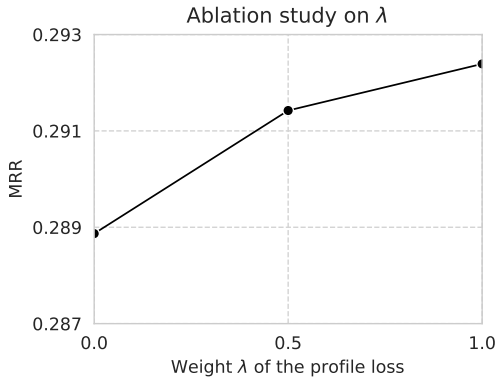


Figure 2: Ablation study on the weight λ of $\mathcal{L}^t_{\text{profile}}$. This confirms the efficacy of our proposed interest profiling.

Efficacy of item interest profiling. To verify the efficacy of our interest profile loss $\mathcal{L}^t_{\text{profile}}$, we conduct an ablation study to investigate the impact of $\mathcal{L}^t_{\text{profile}}$ on the overall recommendation performance of our propsed G2Rec. Specifically, we compare the MRRs under various weights λ on the Beauty dataset. The results are shown in Figure 2. As can be seen from the figure, as the weight λ increases, the recommendation metric MRR also improves. This demonstrates the efficacy of our proposed interest profile loss $\mathcal{L}^t_{\text{profile}}$ in capturing the complex relationships between items and enhancing the accuracy of recommendations. Notably, when $\lambda = 0$ (which means that the interest profile loss is not used), the MRR is significantly lower than when $\lambda > 0$. This suggests that the interest profile loss plays a crucial role in improving the recommendation performance of our proposed G2Rec. Overall, our ablation study demonstrates the importance of the interest profile loss in our proposed G2Rec and highlights its efficacy in improving the quality of recommendations in real-world applications.

6 Related Work

6.1 Generative Recommendation

Generative recommendation formulates recommendation tasks as sequential generation problems, aiming to predict user preferences by generating sequences of recommended items rather than simply ranking predefined candidates [7]. Recent studies have employed deep generative models, particularly autoregressive language models, to directly rank candidate items [15], predict user ratings [9], and retrieve relevant information [70]. P5 [10] conceptualizes recommendation as a language processing task, integrating various recommendation scenarios within a unified sequence-to-sequence framework. This approach effectively leverages multiple sources of information, facilitating richer and more nuanced user modeling. However, bridging the gap between natural language processing and discrete user and item indexing remains challenging. Hua et al. [16] explored several indexing methods, including sequential indexing, collaborative indexing, semantic indexing, and hybrid indexing, to incorporate indexing mechanisms into generative models. More recent studies [22, 57, 60, 61] incorporated both semantic and collaborative information into index tokenization. Despite demonstrating promising outcomes, these approaches continue to face difficulties in fully capturing complex user behavioral patterns and detailed item semantic information.

6.2 Graphs in Recommendation

Graph-based recommendation approaches leverage the inherent relational structure among users and items to capture complex connectivity patterns that traditional methods often overlook. Recent advances in GNNs have propelled significant progress in this area, allowing models to integrate both structural and feature-based information into embedding learning. Notably, Graph Convolutional Matrix Completion (GCMC) [51] introduces a graph autoencoder framework that facilitates differentiable message passing over bipartite user-item graphs. Neural Graph Collaborative Filtering (NGCF) [55] explored learning user and item embeddings by propagating neighbor information in user-item graphs. LightGCN [12] then enhanced this idea by adopting linear neighbor information aggregation. Despite their effectiveness, GNN-based methods typically perceive only a small subgraph centered around individual users or items, limiting their capacity to fully exploit holistic graph information and potentially overlooking valuable global structural insights. More recently, researchers have begun integrating LLMs with GNNs to enhance recommendation performance by leveraging the relational modeling capabilities of GNNs and the natural language understanding strengths of LLMs. Existing approaches generally fall into three categories: graph-augmented LLM [31, 32, 54], LLM-augmented graph [18, 25, 56], and LLM-graph collaboration [47, 69].

7 Conclusion & Future Work

In this paper, we introduced G2Rec, a scalable item schema designed to enhance generative sequential recommendation systems by effectively integrating holistic graph-based co-engagement modeling for semantic tokenization. Our approach addresses the limitations of existing methods, which often struggle with scalability and the accurate incorporation of complex user-behavioral and item-semantic information. By constructing an item-item co-engagement graph and employing a soft clustering algorithm, G2Rec derives interest prototypes that transform user interaction history sequences into the more informative item interest profiles. This transformation allows the recommendation model to capture comprehensive and semantic user interest profiles without relying on predefined ground-truth interests. Our extensive experiments on public datasets, along with successful online deployment on product surfaces, demonstrate the superiority of G2Rec over current state-of-the-art methods. The results highlight that G2Rec have the ability to provide more accurate and holistic modeling of user behavior, ultimately leading to improved recommendation performance in industrial applications. G2Rec represents a significant step forward in the field of generative sequential recommendation, offering a scalable and effective solution for industry-scale applications. Future work may explore further enhancements to the framework, such as incorporating additional data sources or refining the clustering algorithm to capture even more nuanced interests.

Acknowledgments

This work is supported by NSF (2433308). The content of the information in this document does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] Nabiha Asghar. 2016. Yelp dataset challenge: Review rating prediction. *arXiv preprint arXiv:1605.05362* (2016).
[2] Wenxuan Bao, Ruxi Deng, Ruizhong Qiu, Tianxin Wei, Hanghang Tong, and Jingrui He. 2025. Latte: Collaborative test-time adaptation of vision-language models in federated learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.
[3] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of community hierarchies in large networks. *CoRR* abs/0803.0476 (2008). arXiv:0803.0476 http://arxiv.org/abs/0803.0476
[4] Eunice Chan, Zhining Liu, Ruizhong Qiu, Yuheng Zhang, Ross Maciejewski, and Hanghang Tong. 2024. Group Fairness via Group Consensus. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency*. 1788–1808.
[5] Lingjie Chen, Ruizhong Qiu, Siyu Yuan, Zhining Liu, Tianxin Wei, Hyunsik Yoo, Zhichen Zeng, Deqing Yang, and Hanghang Tong. 2024. WAPITI: A watermark for finetuned open-source LLMs.
[6] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).
[7] Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. 2025. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965* (2025).
[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 4171–4186.
[9] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524* (2023).
[10] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM conference on recommender systems*. 299–315.
[11] Xu Guo, Tong Zhang, Yuanzhi Wang, Chenxu Wang, Fuyun Wang, Xudong Wang, Xiaoya Zhang, Xin Liu, and Zhen Cui. 2025. Multi-Modal Hypergraph Enhanced

LLM Learning for Recommendation. *arXiv preprint arXiv:2504.10541* (2025).
[12] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.
[13] Xinyu He, Jian Kang, Ruizhong Qiu, Fei Wang, Jose Sepulveda, and Hanghang Tong. 2024. On the sensitivity of individual fairness: Measures and robust algorithms. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 829–838.
[14] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).
[15] 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*. Springer, 364–381.
[16] Wenyue Hua, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. 2023. How to index item ids for recommendation foundation models. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*. 195–204.
[17] Mike Isaac. 2023. Meta Profit Is Up 16% to $7.8 Billion in Recent Quarter. *The New York Times (Digital Edition)* (2023).
[18] Minhye Jeon, Seokho Ahn, and Young-Duk Seo. 2025. Topic-aware knowledge graph with large language models for interoperability in recommender systems. In *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing*. 795–802.
[19] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.
[20] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.
[21] Ting-Wei Li, Ruizhong Qiu, and Hanghang Tong. 2025. Model-free graph data selection under distribution shift.
[22] Xinhang Li, Chong Chen, Xiangyu Zhao, Yong Zhang, and Chunxiao Xing. 2023. E4SRec: An elegant effective efficient extensible solution of large language models for sequential recommendation. *arXiv preprint arXiv:2312.02443* (2023).
[23] Xiao Lin, Zhining Liu, Dongqi Fu, Ruizhong Qiu, and Hanghang Tong. 2024. BackTime: Backdoor attacks on multivariate time series forecasting. In *Advances in Neural Information Processing Systems*, Vol. 37.
[24] Xiao Lin, Zhining Liu, Ze Yang, Gaotang Li, Ruizhong Qiu, Shuke Wang, Hui Liu, Haotian Li, Sumit Keswani, Vishwa Pardeshi, et al. 2025. MORALISE: A Structured Benchmark for Moral Alignment in Visual Language Models.
[25] Fan Liu, Yaqi Liu, Huilin Chen, Zhiyong Cheng, Liqiang Nie, and Mohan Kankanhalli. 2025. Understanding before recommendation: Semantic aspect-aware review exploitation via large language models. *ACM Transactions on Information Systems* 43, 2 (2025), 1–26.
[26] Lihui Liu, Zihao Wang, Ruizhong Qiu, Yikun Ban, Eunice Chan, Yangqiu Song, Jingrui He, and Hanghang Tong. 2024. Logic query of thoughts: Guiding large language models to answer complex logic queries with knowledge graphs.
[27] Zhining Liu, Ruizhong Qiu, Zhichen Zeng, Hyunsik Yoo, David Zhou, Zhe Xu, Yada Zhu, Kommy Weldemariam, Jingrui He, and Hanghang Tong. 2024. Class-imbalanced graph learning without class rebalancing. In *Proceedings of the 41st International Conference on Machine Learning*.
[28] Zhining Liu, Ruizhong Qiu, Zhichen Zeng, Yada Zhu, Hendrik Hamann, and Hanghang Tong. 2024. AIM: Attributing, interpreting, mitigating data unfairness. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2014–2025.
[29] Zhining Liu, Ze Yang, Xiao Lin, Ruizhong Qiu, Tianxin Wei, Yada Zhu, Hendrik Hamann, Jingrui He, and Hanghang Tong. 2025. Breaking silos: Adaptive model fusion unlocks better time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*.
[30] Zhining Liu, Zhichen Zeng, Ruizhong Qiu, Hyunsik Yoo, David Zhou, Zhe Xu, Yada Zhu, Kommy Weldemariam, Jingrui He, and Hanghang Tong. 2023. Topological augmentation for class-imbalanced node classification.
[31] Luyi Ma, Xiaohan Li, Zezhong Fan, Kai Zhao, Jianpeng Xu, Jason Cho, Praveen Kanumala, Kaushiki Nag, Sushant Kumar, and Kannan Achan. 2024. Triple modality fusion: Aligning visual, textual, and graph data with large language models for multi-behavior recommendations. *arXiv preprint arXiv:2410.12228* (2024).
[32] Qiyao Ma, Xubin Ren, and Chao Huang. 2024. Xrec: Large language models for explainable recommendation. *arXiv preprint arXiv:2406.02377* (2024).
[33] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 43–52.
[34] Mark E. J. Newman. 2006. Modularity and community structure in networks. *Proceedings of the national academy of sciences* 103, 23 (2006), 8577–8582.
[35] Mark E. J. Newman and Michelle Girvan. 2004. Finding and evaluating community structure in networks. *Physical Review E* 69, 2 (2004), 026113.
[36] Andrew Ng, Michael Jordan, and Yair Weiss. 2001. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems* 14 (2001).
[37] Ruizhong Qiu, Jun-Gi Jang, Xiao Lin, Lihui Liu, and Hanghang Tong. 2024. TUCKET: A tensor time series data structure for efficient and accurate factor analysis over time ranges. *Proceedings of the VLDB Endowment* 17, 13 (2024).
[38] Ruizhong Qiu, Gaotang Li, Tianxin Wei, Jingrui He, and Hanghang Tong. 2025. Saffron-1: Safety Inference Scaling.
[39] Ruizhong Qiu, Zhiqing Sun, and Yiming Yang. 2022. DIMES: A differentiable meta solver for combinatorial optimization problems. In *Advances in Neural Information Processing Systems*, Vol. 35. 25531–25546.
[40] Ruizhong Qiu and Hanghang Tong. 2024. Gradient compressed sensing: A query-efficient gradient estimator for high-dimensional zeroth-order optimization. In *Proceedings of the 41st International Conference on Machine Learning*.
[41] Ruizhong Qiu, Dingsu Wang, Lei Ying, H Vincent Poor, Yifang Zhang, and Hanghang Tong. 2023. Reconstructing graph diffusion history from a single snapshot. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1978–1988.
[42] Ruizhong Qiu, Zhe Xu, Wenxuan Bao, and Hanghang Tong. 2025. Ask, and it shall be given: On the Turing completeness of prompting. In *13th International Conference on Learning Representations*.
[43] Ruizhong Qiu, Weiliang Will Zeng, Hanghang Tong, James Ezick, and Christopher Lott. 2025. How efficient is LLM-generated code? A rigorous & high-standard benchmark. In *13th International Conference on Learning Representations*.
[44] Ruiyang Ren, Zhaoyang Liu, Yaliang Li, Wayne Xin Zhao, Hui Wang, Bolin Ding, and Ji-Rong Wen. 2020. Sequential recommendation with self-attentive multi-adversarial network. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 89–98.
[45] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).
[46] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.
[47] Xie Runfeng, Cui Xiangyang, Yan Zhou, Wang Xin, Xuan Zhanwei, Zhang Kai, et al. 2023. Lkpnr: Llm and kg for personalized news recommendation framework. *arXiv preprint arXiv:2308.12028* (2023).
[48] 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*. 1441–1450.
[49] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 565–573.
[50] Vincent A. Traag, Ludo Waltman, and Nees Jan Van Eck. 2019. From Louvain to Leiden: Guaranteeing well-connected communities. *Scientific reports* 9, 1 (2019), 1–12.
[51] Rianne Van Den Berg, N Kipf Thomas, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* 2, 8 (2017), 9.
[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
[53] Dingsu Wang, Yuchen Yan, Ruizhong Qiu, Yada Zhu, Kaiyu Guan, Andrew Margenot, and Hanghang Tong. 2023. Networked time series imputation via position-aware graph enhanced variational autoencoders. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2256–2268.
[54] Xinfeng Wang, Jin Cui, Fumiyo Fukumoto, and Yoshimi Suzuki. 2024. Enhancing high-order interaction awareness in llm-based recommender model. *arXiv preprint arXiv:2409.19979* (2024).
[55] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.
[56] Yan Wang, Zhixuan Chu, Xin Ouyang, Simeng Wang, Hongyan Hao, Yue Shen, Jinjie Gu, Siqiao Xue, James Y Zhang, Qing Cui, et al. 2023. Enhancing recommender systems with large language model reasoning graphs. *arXiv preprint arXiv:2308.10835* (2023).
[57] Ye Wang, Jiahao Xun, Minjie Hong, Jieming Zhu, Tao Jin, Wang Lin, Haoyuan Li, Linjun Li, Yan Xia, Zhou Zhao, et al. 2024. Eager: Two-stream generative recommender with behavior-semantic collaboration. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3245–3254.
[58] Tianxin Wei, Ruizhong Qiu, Yifan Chen, Yunzhe Qi, Jiacheng Lin, Wenju Xu, Sreyashi Nag, Ruirui Li, Hanqing Lu, Zhengyang Wang, Chen Luo, Hui Liu, Suhang Wang, Jingrui He, Qi He, and Xianfeng Tang. 2024. Robust Watermarking for Diffusion Models: A Unified Multi-Dimensional Recipe.
[59] Ziwei Wu, Lecheng Zheng, Yuancheng Yu, Ruizhong Qiu, John Birge, and Jingrui He. 2024. Fair anomaly detection for imbalanced groups.

[60] Longtao Xiao, Haozhao Wang, Cheng Wang, Linfei Ji, Yifan Wang, Jieming Zhu, Zhenhua Dong, Rui Zhang, and Ruixuan Li. 2025. Progressive Collaborative and Semantic Knowledge Fusion for Generative Recommendation. *arXiv preprint arXiv:2502.06269* (2025).

[61] Wujiang Xu, Qitian Wu, Zujie Liang, Jiaojiao Han, Xuying Ning, Yunxiao Shi, Wenfang Lin, and Yongfeng Zhang. 2024. SLMRec: Distilling large language models into small for sequential recommendation. *arXiv preprint arXiv:2405.17890* (2024).

[62] Zhe Xu, Ruizhong Qiu, Yuzhong Chen, Huiyuan Chen, Xiran Fan, Menghai Pan, Zhichen Zeng, Mahashweta Das, and Hanghang Tong. 2024. Discrete-state continuous-time diffusion for graph generation. In *Advances in Neural Information Processing Systems*, Vol. 37.

[63] Hyunsik Yoo, SeongKu Kang, Ruizhong Qiu, Charlie Xu, Fei Wang, and Hanghang Tong. 2025. Embracing plasticity: Balancing stability and plasticity in continual recommender systems. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[64] Hyunsik Yoo, Ruizhong Qiu, Charlie Xu, Fei Wang, and Hanghang Tong. 2025. Generalizable recommender system during temporal popularity distribution shifts. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

[65] Hyunsik Yoo, Zhichen Zeng, Jian Kang, Ruizhong Qiu, David Zhou, Zhining Liu, Fei Wang, Charlie Xu, Eunice Chan, and Hanghang Tong. 2024. Ensuring user-side fairness in dynamic recommender systems. In *Proceedings of the ACM on Web Conference 2024*. 3667–3678.

[66] Kai Yu, Shipeng Yu, and Volker Tresp. 2005. Soft clustering on graphs. *Advances in neural information processing systems* 18 (2005).

[67] Zhichen Zeng, Ruizhong Qiu, Wenxuan Bao, Tianxin Wei, Xiao Lin, Yuchen Yan, Tarek F. Abdelzaher, Jiawei Han, and Hanghang Tong. 2025. Pave your own path: Graph gradual domain adaptation on fused Gromov–Wasserstein geodesics.

[68] Zhichen Zeng, Ruizhong Qiu, Zhe Xu, Zhining Liu, Yuchen Yan, Tianxin Wei, Lei Ying, Jingrui He, and Hanghang Tong. 2024. Graph mixup on approximate Gromov–Wasserstein geodesics. In *Proceedings of the 41st International Conference on Machine Learning*.

[69] Ziwei Zhao, Fake Lin, Xi Zhu, Zhi Zheng, Tong Xu, Shitian Shen, Xueying Li, Zikai Yin, and Enhong Chen. 2024. DynLLM: when large language models meet dynamic graph recommendation. *arXiv preprint arXiv:2405.07580* (2024).

[70] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107* (2023).

[71] Jiaru Zou, Yikun Ban, Zihao Li, Yunzhe Qi, Ruizhong Qiu, Ling Yang, and Jingrui He. 2025. Transformer copilot: Learning from the mistake log in LLM fine-tuning.