

Hypergraph-based Temporal Modelling of Repeated Intent for Sequential Recommendation

Anonymous Author(s)

ABSTRACT

In sequential recommendation scenarios, user intent is a key driver of consumption behavior. However, consumption intents are usually latent and hence, difficult to leverage for recommender systems. Additionally, intents can be of repeated nature (e. g., yearly shopping for christmas gifts or buying a new phone), which has not been exploited by previous approaches. To navigate these impediments we propose the *HyperHawkes* model which models user sessions via hypergraphs and extracts user intents via contrastive clustering. We use Hawkes Processes to model the temporal dynamics of intents, namely repeated consumption patterns and long-term interests of users. For short-term interest adaption, which is more fine-grained than intent-level modeling, we use a multi-level attention mixture network and fuse long-term and short-term signals. We use the generalized expectation-maximization (EM) framework for training the model by alternating between intent representation learning and optimizing parameters of the long- and short-term modules. Extensive experiments on four real-world datasets from different domains show that HyperHawkes significantly outperforms existing state-of-the-art methods.

CCS CONCEPTS

• Information systems → Recommender systems.

KEYWORDS

Recommender Systems, Sequential Recommendation, Graph Neural Network, Hypergraph, Intent

ACM Reference Format:

Anonymous Author(s). 2024. Hypergraph-based Temporal Modelling of Repeated Intent for Sequential Recommendation. In *Proceedings of 2025 ACM Web Conference (WWW '25)*. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/1122445.1122456>

1 INTRODUCTION

Recommender systems have long become essential in filtering information effectively, for instance on video-sharing websites, e-commerce platforms, online bookstores, and social networks. With the abundance of online information, recommender systems have gained increasing importance by discovering and leveraging the underlying (latent) intents of users to cater to their preferences. In

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WWW '25, May 2025, Sidney, Australia

© 2024 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM... \$15.00

<https://doi.org/10.1145/1122445.1122456>

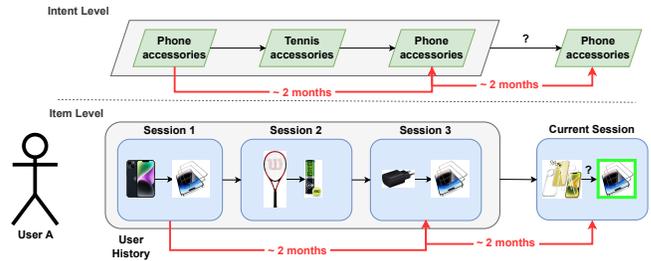


Figure 1: A toy example of an e-commerce retailer scenario with repeated user intents.

recent years, there has been a growing trend in modeling user sequential behaviors, which aims to capture short-term user interest and longer-term sequential patterns including popularity trends and interest drifts [42]. While traditional recommendation methods focus on static user preference modeling [16, 45], Sequential Recommendation (SR) models dynamically characterize user behaviors [18, 24], aiming to accurately predict users' interests in items based on their historical interactions and their corresponding points in time, allowing for more accurate and timely recommendations [8, 54].

The majority of previous works in SR order items by interaction timestamps and focus on sequential patterns to predict the next potential item. Early works adopt Markov chains to provide recommendations based on the L previous interactions via an L -order Markov chain [15, 46]. Also, Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) have been applied to model long- and short-term dependencies in a user interaction sequence [18, 67]. More recent methods rely on the self-attention mechanism and transformer-based models for capturing complex sequential dependencies for next-item recommendations [24, 49]. Another line of work explicitly focuses on modeling temporal dynamics in item sequences based on interaction timestamps [31, 66]. The availability of temporal information also enables models to learn about global events (e. g., Christmas) [56] and the periodicity of items [4, 54]. Previous works in the field model the temporal dynamics on an item level or rely on additional category and knowledge-graph information to represent user intent [19, 55]. However, these approaches come with several downsides: Learning temporal dynamics on the item level is often difficult due to data sparsity and ignores co-occurring item consumption patterns across all users. Also, valuable meta-information for learning user intents is not always available and mostly ignores personal user preferences like preferred brands, price restrictions, or re-consumption behavior.

To fill the aforementioned gaps, we propose to extract latent user intents from the user interaction sequences and model personalized temporal dynamics including example repeat consumption on the user intent level. Consider the example in Figure 1. During each session, User A interacts with the system by e.g., viewing or purchasing items

with different intents, and in this example, their interest is solely focused on the items relevant to their current intent. From the user’s interaction history, it is apparent that the intent of consuming phone accessories is of repeated nature and is connected to the lifetime of a screen protector for the phone. Explicitly modeling this behavior increases the ability to recommend suitable phone accessories after a certain period (e. g., two months).

Repeat consumption occurs due to people’s habits. For instance, we frequently purchase the same items, dine at the same restaurants, and listen to the same songs and artists often with a certain intent [1]. To empirically analyze the intent repeat consumption in the real world, we extract sets of frequently co-occurring item sets via the FP-Max algorithm [13]. For each active user (a user with at least 20 item interactions) we compute maximum frequent item sets (appearing twice or more in the user history) with a size larger than 1 to capture intent-level interactions. Then, we compute the maximum support of all repeated intents per user, where a support of 0.5 of an item set means this intent is apparent in 50% of the user’s sessions. Figure 2 displays the distribution of intent repeat consumption with different maximum support values for four real-world benchmark datasets from different domains. Although there is a large portion of users with non-repeating intents, it is clear to see that intent repeat consumption is prevalent, and also constitutes a significant portion of interactions in certain domains.

To bridge this described gap of modeling temporal dynamics of user intents we propose the **Hypergraph-based Hawkes Processes (HyperHawkes)** model for sequential recommendation. Our approach leverages hypergraphs and soft clustering to extract latent user intent representations from the user interaction data. Based on these user intent representations our temporal excitation module learns the dynamics of user intents and item consumption behavior based on Hawkes Processes [14], a temporal point process to model discrete events in a continuous-time regime. We propose a novel time decay function to represent the excitation strength between historical intent and item behaviors and their corresponding time intervals. To capture short-term interest changes on the item level, we additionally compute short-term interest scores based on an attention mixture network, which captures the influence of the last interacted items in the current session. These steps ensure that our model effectively combines long-term and short-term user interest, and models both intent- and item-level temporal dynamics. We summarize our main technical contributions as follows:

- We propose a novel global item hypergraph construction strategy for learning intent-based item representations and employ soft clustering to extract latent user intents.
- We integrate Hawkes Processes (temporal point processes) to model long-term temporal dynamics on intent level; further, we fuse short-term interests for increased personalized recommendation performance.
- We conduct extensive experiments showing that our proposed model achieves significant performance improvements over a large number of state-of-the-art competitors on four datasets from different domains.¹

¹Code: <https://anonymous.4open.science/r/HyperHawkes-2FB8>

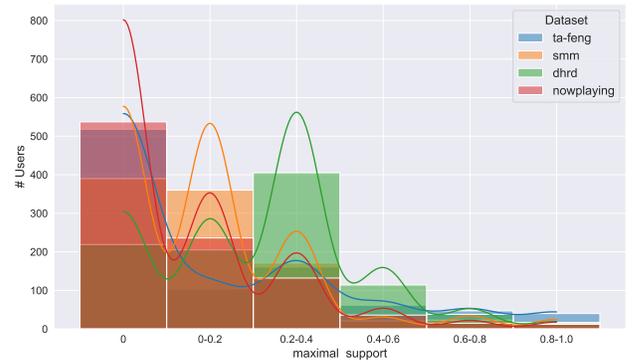


Figure 2: Distribution of maximal support of intents (item sets with size ≥ 2) of active users per dataset. We randomly sampled 1000 users per dataset to ensure comparability between datasets.

2 RELATED WORK

In this section, we review related work, which includes sequential recommendation, user intent, and temporal information learning.

2.1 Sequential Recommendation

Sequential recommendation aims to recommend items to the user by modeling their past behavior sequences and characterize their dynamic interests [24, 39, 42, 46]. Earlier approaches in this field are based on nearest-neighbor methods [12, 21], factorization machine-based methods [44] and Markov Chains [15]. In recent years the advances of deep learning also led to the deployment of many deep sequential recommendation models including CNN-based models [50, 67], RNN-based models [18, 65] and self-attention based models [10, 24, 49]. SASRec [24] and BERT4Rec [49] both utilize the transformer architecture [53] to model correlations among context information in SR. Recently, many works focused on using contrastive self-supervised learning (SSL) to enhance the mutual information between positive samples while increasing the discrimination of negatives [38, 41, 51, 63, 73].

2.2 User Intent for Recommendation

In recent times an increasing body of work studied users’ intents for improving sequential recommendations [29, 30]. Works in session-based recommendation learn different purchase purposes via a mixture-channel purpose routing network [57], use a multi-intent translation graph neural network to mine user intents [35] or employ a dual-intent network to recommend new items [22]. Work in [71] proposes an attention mixture network based on user intents to achieve multi-level reasoning over item transitions. Another area of research focuses on understanding the sequential patterns in users’ interaction behaviors over longer periods. DSSRec [36] introduces a seq2seq training strategy that utilizes multiple future interactions as supervision and incorporates an intent variable derived from both the user’s past and future behavior sequences. In ICLRec [7] user intents are represented by latent variables and learned via clustering. The learned intents are leveraged into SR models via contrastive SSL to

maximize the agreement between the representation of a sequence and its corresponding intent.

2.3 Temporal Information, Repeated Consumption

Time-sensitive recommendation considers the temporal information of item interactions as context features or models temporal decay effects of historical interactions via point processes. In TimeSVD++ [62], timestamps are divided into bins and combined with a collaborative-filtering framework. In tensor factorization methods time is viewed as an extra dimension in the user-item interaction matrix [5, 25, 64]. Other works focus on capturing trends and user-evolving patterns via attention-based temporal modules [8, 9, 43, 66]. Li et al. [31] extend SASRec by modeling the user-specific time intervals in the item sequence. Recently, TGSRec [11] designs a continuous-time bipartite graph, which captures temporal dynamics within the sequential patterns of user-item interactions. Another line of work applies the Hawkes Process framework [14] to model the temporal decay effects of historical interactions [6, 74], which also increases the capability of the model to predict repeating item interactions [4, 19, 54, 55].

Different from previous works, we not only leverage that repeated interactions occur at intent levels but also show that incorporating personal user information is crucial for learning temporal dynamics. Additionally, our model addresses the gaps in the current understanding of user intents, especially in terms of capturing repeated and periodic patterns, modeling user intents through a hypergraph and soft clustering techniques based on user session information, which significantly enhances personalized recommendation performance.

3 PRELIMINARIES

3.1 Problem Definition and Notations

In sequential interaction scenarios, the observed user-item interaction data is represented by a set of tuples $\{(u, v, t)\}$, indicating that user $u \in \mathcal{U}$ interacted with item $v \in \mathcal{V}$ at timestamp t . The interactions are sorted chronologically to form a user’s interaction sequence $I_u^t = [(v_1, t_1), (v_2, t_2), \dots, (v_n, t_n)]$, where n is the number of interactions of user u until timestamp t . Based on the varying time intervals between interactions, the sequence S_u can be divided into subsequences (or sessions) whenever the time interval between two interactions exceeds a threshold δ (e. g., a day or hour). The resulting session interaction sequence can be represented as $S_u = [s_1^u, s_2^u, \dots, s_l^u]$, where s_l^u represents the l -th interaction subsequence of user u containing items from \mathcal{V} . The objective of sequential recommendation is to predict the item from the item set \mathcal{V} that the user u is most likely to interact with at a given timestamp t , given their sequence S_u .

3.2 Hawkes Processes for Sequential Modeling

A temporal point process is a stochastic process consisting of discrete events localized in the continuous-time domain. In sequential recommendation, the times at which a user interacts with a specific item can be represented as a series of historical events $H_t = [t_1, t_2, \dots, t_n]$. To model the time of the next event based on previous events, a conditional intensity function $\lambda(t|H_t)$ is introduced. This function represents a stochastic model for the occurrence of the next event given all previous event times and thereby, affects the characteristics of the temporal point process. In Hawkes Processes [14], the

intensity function takes the form of

$$\lambda(t) = \lambda_{Base} + \alpha \sum_{t_i < t} \varphi(t - t_i), \quad (1)$$

where λ_{Base} represents the base intensity and each historical event has a self-exciting effect on the current intensity controlled by the triggering kernel φ which determines how each past event boosts the event intensity over time. The parameter α determines the degree of excitation. In the context of sequential recommendation, the base intensity represents the user’s basic interest in a target item, and the self-exciting term indicates the cumulative impact of historical interactions on the user’s interest over time.

4 PROPOSED METHOD (HYPERHAWKES)

As illustrated in Figure 3 our HyperHawkes model consists of several major components, including the intent-based global item graph, and a hypergraph-based aggregation layer to generate intent-based item representations for the soft clustering component. The clustered intent-based item representations serve as inputs to the temporal module, which is responsible for capturing users’ long-term interests. To model short-term interest we employ an attention-mixture network and combine both long-term and short-term signals in the final prediction layer. In the following, we will detail each component.

4.1 Intent-based Hypergraph Network

As user intents are latent by definition and hence are difficult to extract, we propose to induce structural bias via hypergraph modeling to support the underlying soft clustering process to find useful intent representations. Compared to a simple graph with an adjacency matrix reflecting the pairwise relationship between two nodes, hyperedges in hypergraphs can connect more than two nodes and are therefore suitable to model user intents, since item interactions on an intent level naturally comprise a set of items. We assume that in each user session, the user interacts with the system based on one or more intents. To build our intent-based global item hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{E} = \{\varepsilon_i\}$ being the set of hyperedges, we apply the following procedure: First, we extract data-driven user intents as frequently occurring item sets across all training user sessions with a length ≥ 2 via the FP-Max algorithm [13], where the minimum support is set to γ . The threshold parameter γ filters for reliable user intents and drops noisy intents not supported by many other user sessions [38]. For each of the extracted intents, we connect all the corresponding items via a hyperedge $\varepsilon_i \in \mathcal{E}$ to build our global hypergraph. Each hyperedge ε_i has a weight w_i attached, indicating the frequency of the extracted intent in the dataset.

To generate intent-based global item representations we design a simple hypergraph aggregation layer. For the item v_i ’s base embedding $\mathbf{x}_i^{(0)}$, we map its corresponding identifier into a dense embedding vector $\mathbf{h}_{v_i} \in \mathbb{R}^d$, where d indicates the dimension. To aggregate information from neighboring nodes we employ the following hypergraph convolution with symmetric normalization in our HGCN component:

$$\mathbf{X}^{(l+1)} = \mathbf{D}^{-1} \mathbf{H} \mathbf{W} \mathbf{B}^{-1} \mathbf{H}^T \mathbf{X}^{(l)}, \quad (2)$$

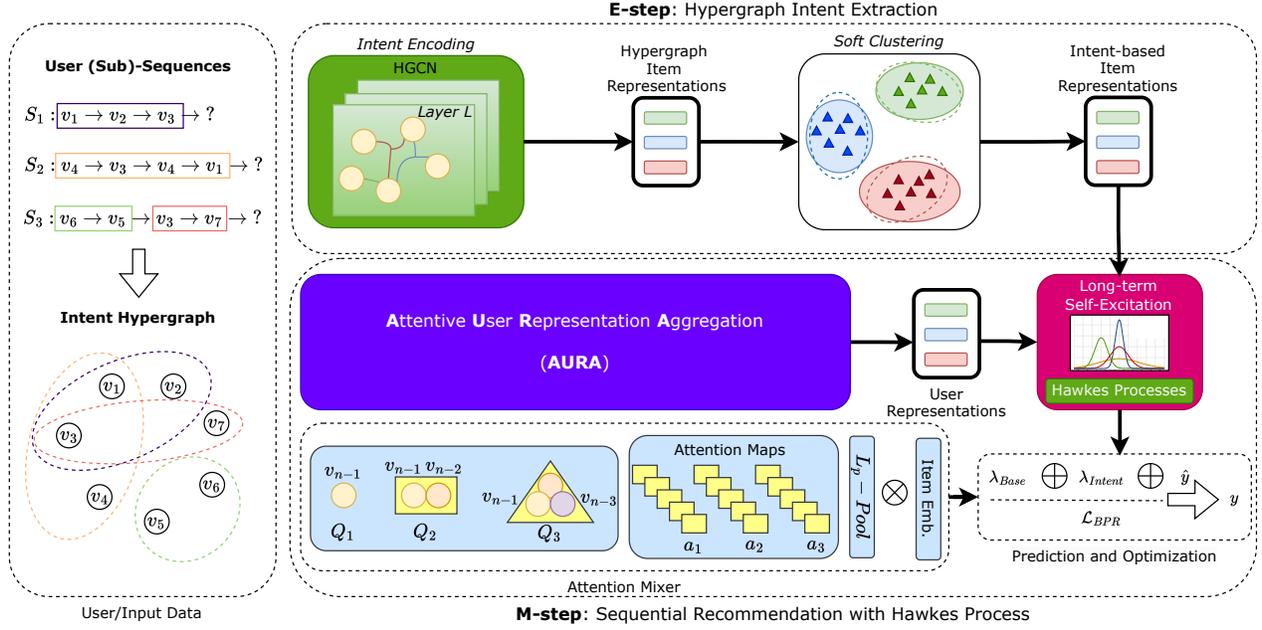


Figure 3: Overall architecture of HyperHawkes: In the E-step of the EM algorithm, our approach extracts latent intent representations via soft clustering of hypergraph-based item embeddings. In the M-step, we compute long-term user preference scores via Hawkes Processes based on the user base excitation from an attentive FISM and self-exciting effects of intents. We fuse short-term scores from the attention-mixture network and the long-term scores to get the preference score of the user for an item.

where \mathbf{H} is the incidence matrix, \mathbf{W} is the diagonal hyperedge weight matrix, and \mathbf{D} and \mathbf{B} are the corresponding degree matrices. Compared to the hypergraph convolution presented in [3] we do not make use of learnable weights and a non-linear activation function, since these components are not essential for recommender systems [59, 61]. To combine node embeddings over multiple layers and increase the receptive field of a node we average the node embeddings over L layers to get the final intent-based hypergraph item representations:

$$\mathbf{X}^{(L)} = \frac{1}{L+1} \sum_{l=0}^L \mathbf{X}^l. \quad (3)$$

4.2 Intent Representation Learning

On the user interaction sequence level, it is easily observed that user sessions exhibit multiple, dynamically shifting intents, where items can also belong to more than one specific intent alone [7, 48]. Additionally, these intents are not confined solely to individual sessions but are also prevalent among users with similar preferences. Therefore, directly utilizing session representation distributions for intent representations will result in a loss of information. To mitigate this, we introduce a soft clustering component to disentangle latent intents and effectively cluster items to intents.

For our soft clustering component we adopt a soft version of the Lloyd’s k -means algorithm [58]. Let \mathbf{x}_j represent the intent-based hypergraph representation $\mathbf{x}_j^{(L)}$ of item v_j and μ_k represent the center of intent cluster k . The variable r_{jk} denotes the probability

to which item v_j is assigned to intent cluster k . In the standard k -means algorithm, this assignment is binary, but we relax it to allow fractional values such that $\sum_k r_{jk} = 1$ for all j . Specifically, we define

$$r_{jk} = \frac{\exp(-\beta \|\mathbf{x}_j - \mu_k\|)}{\sum_{\ell} \exp(-\beta \|\mathbf{x}_j - \mu_{\ell}\|)}, \quad (4)$$

which provides a soft-min assignment of each point to the cluster centers based on distance. We use negative cosine similarity as a distance norm $\|\cdot\|$. Here, β is an inverse-temperature hyperparameter; taking $\beta \rightarrow \infty$ recovers the standard k -means assignment. The intent cluster centers can be optimized via an iterative process similar to the traditional k -means updates by alternately setting

$$\mu_k = \frac{\sum_j r_{jk} \mathbf{x}_j}{\sum_j r_{jk}} \quad \forall k = 1, \dots, K \quad (5)$$

and

$$r_{jk} = \frac{\exp(-\beta \|\mathbf{x}_j - \mu_k\|)}{\sum_{\ell} \exp(-\beta \|\mathbf{x}_j - \mu_{\ell}\|)} \quad \forall k = 1, \dots, K, j = 1, \dots, n. \quad (6)$$

These iterations converge to a fixed point where μ remains unchanged between successive updates. As a result, we have soft intent cluster assignments for each item $\mathbf{p}_j \in \mathbf{P}$ corresponding to probabilities that item v_j belongs to one of the intent clusters K . This probability distribution \mathbf{p}_j serves as the latent intent representation of item v_j .

Since the intent representations $\mathbf{p}_j \in \mathbf{P}$ are latent by definition we face the issue that without the cluster representations, we cannot estimate our model parameters θ and without θ we are not able

to find a result for the soft cluster assignment probabilities \mathbf{P} . It has been shown that a generalized Expectation-Maximization (EM) framework can resolve this situation [7, 34]. In its basic idea, EM starts with an initial guess of θ and estimates the expected values of our cluster assignments \mathbf{P} in the E-step. In the M-step we maximize the objective w.r.t. the model parameters θ given the expected values of \mathbf{P} . These steps are repeated until the likelihood cannot increase anymore. For detailed derivations of the EM framework under the sequential recommendation scenario we refer to [7, 34].

4.3 Repeated Long-term Intent Consumption

We employ Hawkes Processes to model the temporal dynamics of long-term interactions on intent level. As defined in Equation 1 λ_{Base} reflects the long-term base interest of a user in a specific item at a given point in time t , whereas the second part accounts for the self-exciting effects λ_e and can capture repeated intent behaviors. We detail these two components in the following.

4.3.1 User Base Preference. Users often have diverse or even contrastive preferences (e. g., romantic and horror movies). Hence, using a single embedding vector to represent the long-term user interest is a limiting factor [60]. Previous works mitigate this issue by generating a global and non-causal representation of each user interaction sequence. Previous works [23, 33] built the preference representation of a user for an interacted item by a uniform aggregation of the representation of the other items in the interaction sequence. In our approach, we incorporate an attentive user representation aggregation (AURA) to compute the basic strength of the Hawkes Process μ which computes user representations flexibly based on the current target item representation \mathbf{h}_v :

$$\lambda_{Base}(u, v) = \mathbf{h}_u + \sum_{j \in I_u \setminus \{v\}} \frac{\exp(\mathbf{h}_j^\top \mathbf{h}_v)}{\sum_{j' \in I_u \setminus \{v\}} \exp(\mathbf{h}_{j'}^\top \mathbf{h}_v)} \mathbf{h}_v \quad (7)$$

where $\mathbf{h}_u \in \mathbb{R}^d$ defines the latent user representation and is fused with the long-term preference of user u for item v which is a weighted aggregation of the item representations in the user interaction sequence I_u .

4.3.2 Intent Excitation Learning. The trigger kernel of the intensity function in the Hawkes Process captures the changing excitation over time. Our goal is to leverage the time dynamics of a user's next intent and how previous intents can trigger subsequent interactions. The Hawkes Process simulates the time dynamics to predict the probability of the next event. In our approach, we consider interaction events with the same underlying intent for self-excitation. Particularly, we define intent excitation learning as follows:

$$\lambda_{Intent}(u, v, t) = \alpha_k \sum_{(v', t') \in I_u^t} I_K(\mathbf{p}_v, \mathbf{p}_{v'}) \varphi(t - t') \quad (8)$$

where I_K denotes the indicator function which returns 1 if item v and v' belong to the same intent cluster and are in different user sessions, otherwise it returns 0. Since we use a soft clustering approach to assign intent cluster probabilities to each item we use the Kullback–Leibler divergence for finding items that correspond to the same intent clusters:

$$I_K(\mathbf{p}_v, \mathbf{p}_{v'}) = \sum_{x \in \mathcal{X}} \mathbf{p}_v \log \left(\frac{\mathbf{p}_v}{\mathbf{p}_{v'}} \right) > \delta, \quad (9)$$

where δ is a parameter to limit the probability distribution distances per intent cluster assignment. The cluster-related parameter α_k weights the degree of excitation. The temporal kernel function $\varphi(\cdot)$ changes with the time interval $\Delta t = t - t'$ between items of the same intent and is defined as:

$$\varphi(\Delta t) = (1 - \pi_k)E(\Delta t | 1/\beta_k) + \pi_k N(\Delta t | \mu_k, \sigma_k), \quad (10)$$

where we leverage an exponential distribution with intent-based parameter β_k to model short-term intent repeat consumption behavior, which diminishes quickly over time. For long-term repeated behavior we employ a normal distribution with mean μ_k and standard deviation σ_k which are also intent representation-based parameters. Using normal distributions to simulate the user dynamic interest changes captures real-world scenarios like item lifecycles and repeated item consumption behavior [19, 54]. The coefficient π_k balances the two distributions. We learn the corresponding parameters of the distributions $\Theta_{Intent} = \{\alpha_k, \beta_k, \mu_k, \sigma_k, \pi_k\}$ by a non-linear transformation of the user representation \mathbf{h}_u , item representation \mathbf{h}_v and the intent representation \mathbf{p}_v :

$$\Theta_{Intent} = \mathcal{M}(\mathbf{h}_u || \mathbf{h}_v || \mathbf{p}_v), \quad (11)$$

where $\mathcal{M}(\cdot)$ is implemented as a two-layer neural network and $||$ denotes the vector concatenation operation. Compared to previous approaches [19, 54] our distribution parameters are not related to item identifiers, but to the corresponding item and intent representations. Hence, our model learns the temporal dynamics on both, item and intent level, and is able to leverage denser input signals, since the number of intents is usually smaller than the number of items in a dataset. Additionally, the incorporation of the user representation to compute the distribution parameters allows our model to learn user-specific repetition behavior which can vary across intents. For instance, one user buys a new phone including accessories every year whereas another user only buys a new phone if the old one is broken and therefore has a longer intent cycle phase.

We introduced the base intensity $\lambda_{Base}(u, v)$ as well as the long-term self-excitations $\lambda_{Intent}(u, v, t)$ on intent level. Therefore, we define our final long-term excitation for item v_i :

$$\lambda_i(u, v_i, t) = \lambda_{Base}(u, v_i) + \lambda_{Intent}(u, v_i, t) \quad (12)$$

4.4 Attention Mixtures for Short-term User Interest

The aforementioned components model the long-term interest and repeated consumption behavior of a user based on intents. However, a user intent might be of an exploratory nature, or the interest may change dynamically during the session. To capture these dynamics of short-term user interest, we employ an attention mixture mechanism, which has shown to be a promising approach for session-based as well as sequential recommendation [52, 71]. Following [71] we generate multi-level intent queries on the groups of last items in a user interaction sequence with length n by employing the deep sets operation [68] and applying linear transformations per level $m \in \mathcal{M}$:

$$\mathbf{Q}_M = \mathbf{W}_M \left(\sum \{\mathbf{h}_{v_i}\}_{i=n, \dots, n-M+1} \right). \quad (13)$$

These generated queries are then used to compute multi-head attention weights as:

$$\alpha_h = \text{softmax} \left(\frac{\mathbf{Q}\mathbf{W}_Q(\mathbf{K}\mathbf{W}_K)^\top}{\sqrt{d}} \right), \quad (14)$$

where $\mathbf{Q} \in \mathbb{R}^{l \times d}$ is the query matrix, $\mathbf{K} \in \mathbb{R}^{n \times d}$ represents the hidden representation of each item in the sequence and $\mathbf{W}^Q, \mathbf{W}^K \in \mathbb{R}^{d \times d}$ are trainable parameters. We apply L_p -pooling [20] to pool the attention map and multiply the hidden representation of the items in the sequence with the corresponding pooled attention weights to get the final short-term sequence representation \mathbf{s}_u .

4.5 Prediction and Model Optimization

For the next-item prediction task we need to combine long-term and short-term interests of users. We use the short-term sequence representation \mathbf{s}_u to compute the short-term interest score $\hat{y}_i = \mathbf{s}_u^\top \mathbf{h}_{v_i}$, for item v_i . Then, we add the long-term excitation score λ_i and the short-term interest score \hat{y}_i to get the final preference score:

$$y_i = \lambda_i + \hat{y}_i. \quad (15)$$

To learn the parameters of our recommendation model in the M-step of the EM algorithm we adopt the pairwise ranking (BPR) loss for optimization as follows:

$$\mathcal{L}_{BPR} = - \sum_{u \in \mathcal{U}} \sum_{i=1}^{N_u} \log \sigma(y_{ui} - y_{uj}), \quad (16)$$

where σ denotes the sigmoid function and y_{uj} reflects the preference score of user u to a randomly sampled negative item $j \notin I_u^t$.

5 EXPERIMENTS AND RESULTS

In this section, we provide the setup and results of extensive experiments to evaluate our proposed model, where we compare HyperHawkes to various state-of-the-art models in SR. Given our overall goal of investigating the impact of intent repeat-consumption and fusing short- and long-term interests of users, we aim to answer the following research questions:

- **RQ1:** How does our proposed HyperHawkes compare to other state-of-the-art SR methods on different datasets?
- **RQ2:** How do different components in HyperHawkes contribute to the performance?
- **RQ3:** How sensitive is HyperHawkes to different hyperparameter settings (e. g., L , K)?

5.1 Experimental Setup

5.1.1 Datasets and Preprocessing. To evaluate the performance of our approach, we conduct experiments on four representative datasets from the e-commerce, food delivery, and music domains [17, 27]. *Ta-Feng*² is a dataset containing Chinese grocery store transaction data from 2001. *SMM*³ chronicles user behavior captured over the span of five months from January 15 to May 15, 2023 from a large online store [47]. For this industrial-scale dataset, we sample 20,000 random users to maintain consistency with the other datasets. The *DHRD* (Delivery Hero Recommendation Dataset)⁴ is

²<https://www.kaggle.com/retailrocket/e-commerce-dataset>

³https://disk.yandex.ru/d/fSEBIQYZusAAuw/datasets/data_smm

⁴<https://github.com/deliveryhero/dh-reco-dataset>

a dataset presented in [2] and comprises food delivery orders from three distinct cities, encompassing different vendors and dishes. For our evaluation, we use the data related to the city of Stockholm. Lastly, the *NowPlaying* dataset comes from the music domain and is described in [70]. This dataset includes music listening behavior of users based on Twitter data. It is worth noting, that we do not provide evaluation for the widely used Amazon review datasets [37], the MovieLens datasets⁵, or the Yelp review datasets⁶, since those datasets are rating/review-based and therefore do not include repeated item consumptions, making them unsuitable for the scenario of repeated intent modeling [17, 27].

Table 1: Dataset statistics (after preprocessing): Number of users, items, interactions, average sequence length and sparsity.

Dataset	Ta-Feng	SMM	DHRD	NowPlaying
$ \mathcal{U} $	26,162	12,098	42,774	11,310
$ \mathcal{V} $	15,642	22,167	20,883	15,905
# Interactions	0.78m	0.87m	0.52m	1.12m
Avg. length	29.99	71.97	12.30	86.39
Sparsity	99.80%	99.67%	99.94%	99.45%

We follow the preprocessing steps as shown in [7, 19] for the four datasets. To be more specific, we keep the *5-core* datasets, where users and items with less than 5 interactions are filtered out. Table 1 provides an overview of the datasets after preprocessing. To split the datasets, we follow common practice in sequential recommendation and use interactions with the second latest time for validation and interactions with the latest timestamp for testing.

5.1.2 Evaluation Metrics. Following previous works [17, 28, 69], we use the whole item set without negative sampling to rank the predictions. We adopt $HR@{5,20}$ (Hit Ratio) and $NDCG@{5,20}$ (Normalized Discounted Cumulative Gain) to evaluate the quality of the recommendation results.

5.1.3 Baseline Methods. We compare HyperHawkes with the following representative baseline and state-of-the-art methods for sequential recommendation:

Static models: BPR-MF [45] is a non-sequential model and characterizes the pairwise interactions via matrix factorization.

Standard sequential & Transformer models: As standard sequential models we include GRU4Rec [18], an RNN-based method and SASRec [24] as transformer-based baseline method for SR.

Temporal & intent-based models: SLRC [54] is a widely used model and one of the first to model item repeat consumption. It combines matrix factorization with a temporal point process, effectively capturing short-term and product lifetime effects. RepeatNet [43] proposes a novel repeat-explore mechanism to balance repeated and new item consumption. For intent-based methods, we include HIDE [32] which models intents via session hypergraphs. Other state-of-the-art approaches include ICLRec [7] and ICSRec [40], where user intents are learned via clustering and Atten-Mixer [71], where intents are modelled via a multi-level network.

⁵<https://grouplens.org/datasets/movielens>

⁶<https://www.yelp.com/dataset>

Table 2: Model performance on all four datasets (\pm standard deviation for HyperHawkes). All improvements of HyperHawkes over the second best model are significant (paired t -test, $p < .05$), best results are in boldface and the second-best results are underlined.

Dataset	Metric	BPR-MF	GRU4Rec	SASRec	SLRC	RepeatNet	HIDE	ICLRec	Atten-Mixer	ICSRec	HyperHawkes	Improv.
Ta-Feng	HR@5	0.0699	0.0657	0.0812	0.0714	0.0432	0.0616	0.0746	<u>0.0878</u>	0.0784	0.1108 \pm 0.0015	26.19%
	HR@20	0.0943	0.1215	0.1629	0.1284	0.1006	0.0853	0.1415	<u>0.1645</u>	0.1566	0.1984 \pm 0.0030	20.60%
	NDCG@5	0.0541	0.0459	0.0528	0.0488	0.0307	0.0419	0.0527	<u>0.0605</u>	0.0519	0.0765 \pm 0.0014	26.44%
	NDCG@20	0.0610	0.0616	0.0761	0.0650	0.0469	0.0485	0.0716	<u>0.0823</u>	0.0742	0.1015 \pm 0.0015	23.32%
SMM	HR@5	0.0542	0.0586	0.0876	0.1170	<u>0.1291</u>	0.0391	0.0526	0.0817	0.0686	0.1483 \pm 0.0019	14.87%
	HR@20	0.1056	0.1323	0.1687	0.1853	<u>0.1968</u>	0.0781	0.1101	0.1638	0.1505	0.2444 \pm 0.0009	24.19%
	NDCG@5	0.0373	0.0393	0.0606	0.0840	<u>0.0972</u>	0.0272	0.0357	0.0561	0.0427	0.1018 \pm 0.0002	4.73%
	NDCG@20	0.0516	0.0602	0.0836	0.1037	<u>0.1175</u>	0.0383	0.0520	0.0793	0.0656	0.1294 \pm 0.0004	10.13%
DHRD	HR@5	0.2156	0.1439	0.2065	<u>0.2775</u>	0.2702	0.1878	0.2554	0.2211	0.2129	0.2982 \pm 0.0055	7.45%
	HR@20	0.3805	0.3214	0.4651	0.4158	0.3211	0.2625	0.4544	0.4295	<u>0.4715</u>	0.4830 \pm 0.0019	2.43%
	NDCG@5	0.1488	0.0946	0.1303	<u>0.2031</u>	0.1983	0.1356	0.1740	0.1489	0.1323	0.2089 \pm 0.0031	2.85%
	NDCG@20	0.1963	0.1450	0.2039	<u>0.2430</u>	0.2142	0.1572	0.2312	0.2084	0.2145	0.2621 \pm 0.0069	7.86%
NowPlaying	HR@5	0.1272	0.0992	0.1229	0.1756	<u>0.1765</u>	0.1079	0.1654	0.1475	0.1375	0.1842 \pm 0.0011	4.36%
	HR@20	0.2730	0.2327	0.2715	0.3117	0.2996	0.1984	<u>0.3135</u>	0.3011	0.2931	0.3526 \pm 0.0006	12.47%
	NDCG@5	0.0879	0.0650	0.0802	0.1197	<u>0.1217</u>	0.0787	0.1156	0.1028	0.0929	0.1242 \pm 0.0007	2.05%
	NDCG@20	0.1289	0.1025	0.1221	<u>0.1589</u>	0.1581	0.1042	0.1574	0.1462	0.1368	0.1713 \pm 0.0008	8.43%

5.1.4 Implementation Details. For a fair comparison, we rely on the RecBole framework [72] to implement our approach, using the provided implementations of the baseline models or re-implementing them accordingly. For all baseline models and our model, the embedding size is set to 64 and the batch size to 512. We do not limit the number of training epochs per model, but adopt an early-stopping strategy, which stops training after five consecutive rounds of performance decrease on the validation set. Each baseline model is optimized according to its corresponding hyperparameters.

For the optimization of HyperHawkes, we use Adam [26] with a learning rate of 0.001. The number of layers L in the HGCN component and number of intent clusters K are searched in the ranges of $\{1, 2, 3, 4, 5\}$ and $\{2, 4, \dots, 128\}$ respectively. For the attention mixture network, we search the number of heads in the range of $\{1, 2, 4, 8\}$ and the number of levels M in $\{1, 2, \dots, 10\}$. The threshold parameters γ and δ are set to $5e-4$ and $1e-12$, correspondingly. Our implementation is based on PyTorch 1.13.1 and Python 3.8.16. All experiments are performed on a workstation with an AMD Ryzen 2950X, a GeForce RTX 2070, and 256 GB main memory. We publish the code and the pre-processed datasets on GitHub⁷.

5.2 Performance Comparison (RQ1)

In Table 2 we report the results of the the performance comparison of HyperHawkes and the proposed baselines. Surprisingly, the results do not affirm that sequential models generally outperform non-sequential methods since BPR-MF shows competing performance compared to GRU4Rec or SASRec. This displays the importance of learning temporal dynamics of repeated user behavior and the incorporation of user intent.

Advanced time-sensitive sequential models often incorporate additional temporal signals to augment recommendation performance. For instance, TiSASRec integrates both the item positions and time intervals in a sequence, yielding superior performance than its transformer-based counterpart SASRec. We further observe that

leveraging contrastive SSL in transformer-based architectures can improve performance, as exhibited by ICLRec which optimizes sequence representations via contrastive SSL at the user intent level. Also, the other intent-based method Atten-Mixer shows significant performance gains over standard sequential models. Among the baseline methods, SLRC and RepeatNet exhibit improved performance even over more sophisticated temporal and intent-based models, underpinning their robustness in recommendation tasks and their ability to model item repeat consumption.

Our model, HyperHawkes, triumphs over all other methods across all datasets, marking a significant advancement in the domain. The average improvements compared with the best baseline per dataset range from 2.43% to 24.19% in HR@20 and from 7.86% to 23.32% in NDCG@20. We attribute this increase in performance to the ability of our approach to effectively model long-term intent repeat behavior and short-term user interest, which we show in detail in our ablation study.

In terms of efficiency and model complexity, we report the training time per epoch on the *Ta-Feng* dataset as a practical proxy for model complexity. Intent-based models like HIDE, ICLRec, Atten-Mixer and ICSRec require 2231.63, 254.21, 13.10 and 174.17 seconds/epoch, respectively. SLRC and RepeatNet, which focus on repeat consumption, need 13.31s and 29.64s, correspondingly. Our HyperHawkes takes 27.75s per epoch on training and therefore, is more efficient than most of the other sequential neural network models, while substantially outperforming these models in recommendation performance. A similar trend in model complexity is also seen for the other datasets.

5.3 Ablation Study (RQ2)

HyperHawkes contains several components including a hypergraph-based graph convolutional network (HGCN), soft clustering (SC), user base interest (LT-UE), intent excitation learning (LT-SINE), and a short-term attention mixture network (ST-ATM). To verify the effectiveness of each component, we conduct an ablation study

⁷<https://anonymous.4open.science/r/HyperHawkes-2FB8>

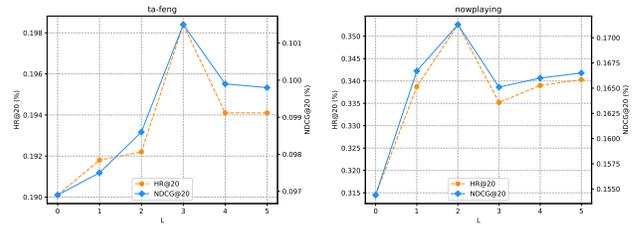
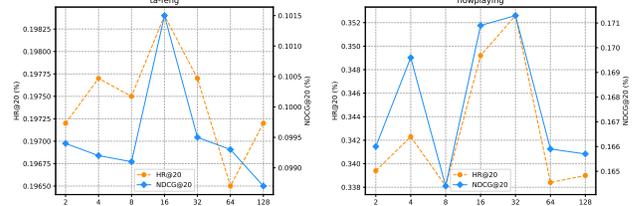
Table 3: Ablation study of HyperHawkes. The symbol ↓ indicates a performance drop of more than 10%, ND = NDCG.

Model	Dataset	Ta-Feng		NowPlaying	
		HR@20	ND@20	HR@20	ND@20
(A) w/o LT-SINE		0.1632↓	0.0842↓	0.3331	0.1637
(B) w/o LT-UE		0.1818	0.0911↓	0.3241	0.1570
(C) w/o HGCN		0.1901	0.0969	0.3145↓	0.1544↓
(D) w/o SC		0.1732↓	0.0867↓	0.3377	0.1666
(E) only ST-ATM		0.1668↓	0.0841↓	0.2954↓	0.1451↓
(F) w/o ST-ATM		0.0914↓	0.0558↓	0.3314	0.1625
HyperHawkes		0.1984	0.1015	0.3526	0.1713

on two datasets. Table 3 displays the result of the ablation study on the *Ta-Feng* and *NowPlaying* datasets. These two datasets were chosen due to their different domains and characteristics in terms of repeat consumption (e. g., e-commerce vs. music streaming). From (A) and (B) we can see the impact of different components in the Hawkes Process for modeling temporal dynamics. Eliminating the intent excitation learning (A) or the user base preference (B), each notably diminishes the performance of the recommendation model to a similar extent. This shows the importance of extracting latent intents and modeling repeat behavior on the intent level compared to the item level only. We also investigate the effect of our proposed hypergraph-based network in (C), where removing the component also leads to a significant performance drop. This backs our assumption that inducing structural bias through the HGCN supports the soft clustering process and leads to more representative cluster/intent representations. Similar effects can be observed when dropping the soft clustering component in (D) and using a standard k -means instead, which showcases the benefit of disentangling user intents via soft probability distributions. Lastly, we explore the effects of the short-term attention mixture network. Relying only on the short-term component without any consideration of long-term effects (E) results in a noticeable performance drop. Dropping the short-term component (F) from HyperHawkes shows a substantial decline compared to the full model, reflecting the critical role of short-term user behavior understanding. The incorporation of both short-term and long-term effects leads to the best overall performance. The ablation study results for the other two datasets *SMM* and *DHRD* are consistent with these findings, but are not reported due to space constraints.

5.4 Impact of Hyper-Parameters (RQ3)

In this section, we investigate the impact of different hyper-parameters. We focus on the number of layers L in the HGCN and the number of intent clusters K , since these hyper-parameters are related to the intent excitation learning, which has shown to have the highest impact on the performance of the final model (see Section 5.3). Figure 4a shows the performance of our model with different settings of layers L on the *Ta-Feng* and *NowPlaying* datasets. A higher number of layers in the hypergraph-based network does not necessarily lead to an increase in performance due to oversmoothing, where node representations converge to the same values. We can find a sweet spot layer setting L of 3 (*Ta-Feng*) and 2 (*NowPlaying*).

**(a) Different number of layers L in our HGCN component.****(b) Number of intent clusters parameter K .****Figure 4: Impact of hyper-parameters in HyperHawkes.**

Our main contribution comprises the temporal modeling of user intents, where those user intents are extracted by soft clustering. Therefore, we have to define the number of clusters K before the training of the model. Given the heterogeneity of datasets, this hyper-parameter needs to be carefully tuned to extract useful intent representations, or in other words, K needs to be tailored to the characteristics of each dataset individually. Figure 4b shows the performance differences between runs with a different number of intent clusters. As the datasets stem from completely different domains, their best-performing setting also vary. For *Ta-Feng* its 16 clusters and for *NowPlaying* the best setting is 32 clusters.

6 CONCLUSION

In this paper, we propose HyperHawkes, a novel Hypergraph-based Hawkes Process model to comprehensively model temporal dynamics of user intents for generating personalized sequential recommendations. We extract intent representations via soft clustering of hypergraph-based item representations. Our model learns the long-term excitation of intents and items via Hawkes Processes and models short-term interests of users via a custom attention mixture component. The corresponding user preference scores from the long-term and short-term components are fused to provide temporal and personalized recommendations. The steps of finding clusters and learning temporal dynamics are alternately optimized under a generalized EM framework. Our extensive experimental results on four real-world datasets demonstrate the effectiveness of our proposed model over state-of-the-art methods. Our approach outperforms all other state-of-the-art methods, which only model repeated consumption on item level or use intents for contrastive learning purposes, in each of the provided metrics. The ablation study showed the impact of each component and that modeling repeat consumption is more important than focusing on short-term interest shifts of users. In future work, we aim to explore the temporal aspects of the extracted user intents for explainability purposes.

REFERENCES

- [1] Ashton Anderson, Ravi Kumar, Andrew Tomkins, and Sergei Vassilvitskii. 2014. The dynamics of repeat consumption. In *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014*. ACM, 419–430.
- [2] Yernat Assylbekov, Raghav Bali, Luke Bovard, and Christian Klau. 2023. Delivery Hero Recommendation Dataset: A Novel Dataset for Benchmarking Recommendation Algorithms. In *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*. ACM, 1042–1044.
- [3] Song Bai, Feihu Zhang, and Philip H. S. Torr. 2021. Hypergraph convolution and hypergraph attention. *Pattern Recognit.* 110 (2021), 107637.
- [4] Ting Bai, Lixin Zou, Wayne Xin Zhao, Pan Du, Weidong Liu, Jian-Yun Nie, and Ji-Rong Wen. 2019. CTRec: A Long-Short Demands Evolution Model for Continuous-Time Recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*. ACM, 675–684.
- [5] Preeti Bhargava, Thomas Phan, Jiayu Zhou, and Juhan Lee. 2015. Who, What, When, and Where: Multi-Dimensional Collaborative Recommendations Using Tensor Factorization on Sparse User-Generated Data. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, Aldo Gangemi, Stefano Leonardi, and Alessandro Panconesi (Eds.). ACM, 130–140.
- [6] Renjin Cai, Xueying Bai, Zhenrui Wang, Yuling Shi, Parikshit Sondhi, and Hongning Wang. 2018. Modeling Sequential Online Interactive Behaviors with Temporal Point Process. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*. ACM, 873–882.
- [7] Yongjun Chen, Zhiwei Liu, Jia Li, Julian J. McAuley, and Caiming Xiong. 2022. Intent Contrastive Learning for Sequential Recommendation. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*. ACM, 2172–2182.
- [8] Junsu Cho, Dongmin Hyun, SeongKu Kang, and Hwanjo Yu. 2021. Learning Heterogeneous Temporal Patterns of User Preference for Timely Recommendation. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*. ACM / IW3C2, 1274–1283.
- [9] Sung Min Cho, Eunhyeok Park, and Sungjoo Yoo. 2020. MEANTIME: Mixture of Attention Mechanisms with Multi-temporal Embeddings for Sequential Recommendation. In *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020*. ACM, 515–520.
- [10] Gabriel de Souza Pereira Moreira, Sara Rabhi, Jeong Min Lee, Ronay Ak, and Even Oldridge. 2021. Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation. In *RecSys '21: Fifteenth ACM Conference on Recommender Systems*. ACM, 143–153.
- [11] Ziwei Fan, Zhiwei Liu, Jiawei Zhang, Yun Xiong, Lei Zheng, and Philip S. Yu. 2021. Continuous-Time Sequential Recommendation with Temporal Graph Collaborative Transformer. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*. ACM, 433–442.
- [12] Diksha Garg, Priyanka Gupta, Pankaj Malhotra, Lovkesh Vig, and Gautam Shroff. 2019. Sequence and Time Aware Neighborhood for Session-based Recommendations: STAN. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019*. ACM, 1069–1072.
- [13] Gösta Grahne and Jianfei Zhu. 2003. Efficiently Using Prefix-trees in Mining Frequent Itemsets. In *FIMI '03, Frequent Itemset Mining Implementations, Proceedings of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations, 19 December 2003, Melbourne, Florida, USA (CEUR Workshop Proceedings, Vol. 90)*. CEUR-WS.org.
- [14] Alan G Hawkes. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika* 58, 1 (1971), 83–90.
- [15] Ruining He and Julian J. McAuley. 2016. Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation. In *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain*. IEEE Computer Society, 191–200.
- [16] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*. ACM, 173–182. <https://doi.org/10.1145/3038912.3052569>
- [17] Balázs Hidasi and Ádám Tibor Czapp. 2023. Widespread Flaws in Offline Evaluation of Recommender Systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 848–855.
- [18] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *4th International Conference on Learning Representations, ICLR 2016, Conference Track Proceedings*.
- [19] Chengkai Huang, Shoujin Wang, Xianzhi Wang, and Lina Yao. 2023. Modeling Temporal Positive and Negative Excitation for Sequential Recommendation. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, Ying Ding, Jie Tang, Juan F. Sequeda, Lora Aroyo, Carlos Castillo, and Geert-Jan Houben (Eds.). ACM, 1252–1263.
- [20] Aapo Hyvärinen and Urs Köster. 2007. Complex cell pooling and the statistics of natural images. *Network: Computation in Neural Systems* 18, 2 (2007), 81–100.
- [21] Dietmar Jannach and Malte Ludewig. 2017. When Recurrent Neural Networks meet the Neighborhood for Session-Based Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017*. ACM, 306–310.
- [22] Di Jin, Luzhi Wang, Yizhen Zheng, Guojie Song, Fei Jiang, Xiang Li, Wei Lin, and Shirui Pan. 2023. Dual Intent Enhanced Graph Neural Network for Session-based New Item Recommendation. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*. ACM, 684–693.
- [23] Santosh Kabbur, Xia Ning, and George Karypis. 2013. FISM: factored item similarity models for top-N recommender systems. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*. ACM, 659–667.
- [24] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *IEEE International Conference on Data Mining, ICDM 2018*. IEEE Computer Society, 197–206.
- [25] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. 2010. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010*, Xavier Amatriain, Marc Torrens, Paul Resnick, and Markus Zanker (Eds.). ACM, 79–86.
- [26] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. <http://arxiv.org/abs/1412.6980>
- [27] Anton Klenitskiy, Anna Volodkevich, Anton Pembek, and Alexey Vasilev. 2024. Does It Look Sequential? An Analysis of Datasets for Evaluation of Sequential Recommendations. In *Proceedings of the 18th ACM Conference on Recommender Systems (Bari, Italy) (RecSys '24)*. Association for Computing Machinery, New York, NY, USA, 1067–1072.
- [28] Walid Krichene and Steffen Rendle. 2020. On Sampled Metrics for Item Recommendation. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (Eds.). ACM, 1748–1757.
- [29] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 2615–2623.
- [30] Haoyang Li, Xin Wang, Ziwei Zhang, Jianxin Ma, Peng Cui, and Wenwu Zhu. 2022. Intention-Aware Sequential Recommendation With Structured Intent Transition. *IEEE Trans. Knowl. Data Eng.* 34, 11 (2022), 5403–5414.
- [31] Jiacheng Li, Yujie Wang, and Julian J. McAuley. 2020. Time Interval Aware Self-Attention for Sequential Recommendation. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*. ACM, 322–330.
- [32] Yinfeng Li, Chen Gao, Hengliang Luo, Depeng Jin, and Yong Li. 2022. Enhancing Hypergraph Neural Networks with Intent Disentanglement for Session-based Recommendation. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*. ACM, 1997–2002. <https://doi.org/10.1145/3477495.3531794>
- [33] Jing Lin, Weike Pan, and Zhong Ming. 2020. FISSA: Fusing Item Similarity Models with Self-Attention Networks for Sequential Recommendation. In *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020*. ACM, 130–139.
- [34] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving Graph Collaborative Filtering with Neighborhood-enriched Contrastive Learning. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*. ACM, 2320–2329.
- [35] Zhiwei Liu, Xiaohan Li, Ziwei Fan, Stephen Guo, Kannan Achan, and Philip S. Yu. 2020. Basket Recommendation with Multi-Intent Translation Graph Neural Network. In *2020 IEEE International Conference on Big Data (IEEE BigData 2020), Atlanta, GA, USA, December 10-13, 2020*. IEEE, 728–737.
- [36] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled Self-Supervision in Sequential Recommenders. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*. ACM, 483–491.
- [37] Julian J. 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, Santiago, Chile, August 9-13, 2015*. ACM, 43–52.

- 1045 [38] Andreas Peintner, Amir Reza Mohammadi, and Eva Zangerle. 2023. SPARE: Shortest Path Global Item Relations for Efficient Session-based Recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*. ACM, 58–69. 1046
- 1047 [39] Andreas Peintner, Marta Moscati, Emilia Parada-Cabaleiro, Markus Schedl, and Eva Zangerle. 2022. Unsupervised Graph Embeddings for Session-based Recommendation with Item Features. In *CARS: Workshop on Context-Aware Recommender Systems (RecSys '22)*. 1048
- 1049 [40] Xiuyuan Qin, Huanhuan Yuan, Pengpeng Zhao, Guanfeng Liu, Fuzhen Zhuang, and Victor S. Sheng. 2024. Intent Contrastive Learning with Cross Subsequences for Sequential Recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining, WSDM 2024, Merida, Mexico, March 4-8, 2024*. ACM, 548–556. <https://doi.org/10.1145/3616855.3635773> 1050
- 1051 [41] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive Learning for Representation Degeneration Problem in Sequential Recommendation. In *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Tempe, AZ, USA, February 21 - 25, 2022*. ACM, 813–823. 1052
- 1053 [42] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-aware recommender systems. *ACM Computing Surveys (CSUR)* 51, 4 (2018), 1–36. 1054
- 1055 [43] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2019. RepeatNet: A Repeat Aware Neural Recommendation Machine for Session-Based Recommendation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 4806–4813. 1056
- 1057 [44] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International conference on data mining*. IEEE, 995–1000. 1058
- 1059 [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). 1060
- 1061 [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, WWW, 2010, ACM*, 811–820. 1062
- 1063 [47] Valeriy Shevchenko, Nikita Belousov, Alexey Vasilev, Vladimir Zholobov, Artyom Sosodka, Natalia Semenova, Anna Volodkevich, Andrey Savchenko, and Alexey Zaytsev. 2024. From Variability to Stability: Advancing RecSys Benchmarking Practices. *arXiv preprint arXiv:2402.09766* (2024). 1064
- 1065 [48] Jiajie Su, Chaochao Chen, Weiming Liu, Fei Wu, Xiaolin Zheng, and Haoming Lyu. 2023. Enhancing hierarchy-aware graph networks with deep dual clustering for session-based recommendation. In *Proceedings of the ACM Web Conference 2023*. 165–176. 1066
- 1067 [49] 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, CIKM 2019, ACM*, 1441–1450. 1068
- 1069 [50] Jiayi 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, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*. ACM, 565–573. <https://doi.org/10.1145/3159652.3159656> 1070
- 1071 [51] Changxin Tian, Zihan Lin, Shuqing Bian, Jinpeng Wang, and Wayne Xin Zhao. 2022. Temporal Contrastive Pre-Training for Sequential Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*. ACM, 1925–1934. 1072
- 1073 [52] Viet-Anh Tran, Guillaume Salha-Galvan, Bruno Sguerra, and Romain Hennequin. 2023. Attention Mixtures for Time-Aware Sequential Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*. ACM, 1821–1826. 1074
- 1075 [53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 5998–6008. <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html> 1076
- 1077 [54] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2019. Modeling Item-Specific Temporal Dynamics of Repeat Consumption for Recommender Systems. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*. ACM, 1977–1987. 1078
- 1079 [55] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Make It a Chorus: Knowledge- and Time-aware Item Modeling for Sequential Recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. ACM, 109–118. 1080
- 1081 [56] Jianling Wang, Raphael Louca, Diane Hu, Caitlin Cellier, James Caverlee, and Liangjie Hong. 2020. Time to Shop for Valentine's Day: Shopping Occasions and Sequential Recommendation in E-commerce. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*. James Caverlee, Xia (Ben) Hu, Mounia Lalmas, and Wei Wang (Eds.). ACM, 645–653. <https://doi.org/10.1145/3336191.3371836> 1103
- 1082 [57] Shoujin Wang, Liang Hu, Yan Wang, Quan Z Sheng, Mehmet Orgun, and Longbing Cao. 2019. Modeling multi-purpose sessions for next-item recommendations via mixture-channel purpose routing networks. In *International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence. 1104
- 1083 [58] Bryan Wilder, Eric Ewing, Bistra Dilkina, and Milind Tambe. 2019. End to end learning and optimization on graphs. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. 4674–4685. 1105
- 1084 [59] Felix Wu, Amauri H. Souza Jr., Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. 2019. Simplifying Graph Convolutional Networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA (Proceedings of Machine Learning Research, Vol. 97)*. PMLR, 6861–6871. 1106
- 1085 [60] Liwei Wu, Shuqing Li, Cho-Jui Hsieh, and James Sharpnack. 2020. SSE-PT: Sequential Recommendation Via Personalized Transformer. In *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020*. ACM, 328–337. 1107
- 1086 [61] Xin Xia, Hongzhi Yin, Junliang Yu, Yingxia Shao, and Lizhen Cui. 2021. Self-Supervised Graph Co-Training for Session-based Recommendation. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management*. ACM, 2180–2190. 1108
- 1087 [62] Liang Xiang and Qing Yang. 2009. Time-Dependent Models in Collaborative Filtering Based Recommender System. In *2009 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2009, Milan, Italy, 15-18 September 2009, Main Conference Proceedings*. IEEE Computer Society, 450–457. 1109
- 1088 [63] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive Learning for Sequential Recommendation. In *38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022*. IEEE, 1259–1273. 1110
- 1089 [64] Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff G. Schneider, and Jaime G. Carbonell. 2010. Temporal Collaborative Filtering with Bayesian Probabilistic Tensor Factorization. In *Proceedings of the SIAM International Conference on Data Mining, SDM 2010, April 29 - May 1, 2010, Columbus, Ohio, USA*. SIAM, 211–222. 1111
- 1090 [65] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Jiajie Xu, Victor S. Sheng, Zhiming Cui, Xiaofang Zhou, and Hui Xiong. 2019. Recurrent Convolutional Neural Network for Sequential Recommendation. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*. ACM, 3398–3404. 1112
- 1091 [66] Wenwen Ye, Shuaiqiang Wang, Xu Chen, Xuepeng Wang, Zheng Qin, and Dawei Yin. 2020. Time Matters: Sequential Recommendation with Complex Temporal Information. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. ACM, 1459–1468. 1113
- 1092 [67] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M. Jose, and Xiangnan He. 2019. A Simple Convolutional Generative Network for Next Item Recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*. ACM, 582–590. <https://doi.org/10.1145/3289600.3290975> 1114
- 1093 [68] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. 2017. Deep Sets. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 3391–3401. 1115
- 1094 [69] Eva Zangerle and Christine Bauer. 2023. Evaluating Recommender Systems: Survey and Framework. *ACM Comput. Surv.* 55, 8 (2023), 170:1–170:38. 1116
- 1095 [70] Eva Zangerle, Martin Pichl, Wolfgang Gassler, and Günther Specht. 2014. #now-playing Music Dataset: Extracting Listening Behavior from Twitter. In *Proceedings of the First International Workshop on Internet-Scale Multimedia Management, WISMM '14, Orlando, Florida, USA, November 7, 2014*. ACM, 21–26. 1117
- 1096 [71] Peiyan Zhang, Jiayan Guo, Chaozhuo Li, Yueqi Xie, Jaeboum Kim, Yan Zhang, Xing Xie, Haohan Wang, and Sunghun Kim. 2023. Efficiently Leveraging Multi-level User Intent for Session-based Recommendation via Atten-Mixer Network. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, WSDM 2023, Singapore, 27 February 2023 - 3 March 2023*. ACM, 168–176. 1118
- 1097 [72] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyuan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management*, 1119

