# $S^2$HyRec: Self-Supervised Hypergraph Sequential Recommendation

**Yuchen Liu[1], Kunyu Ni[1], Zhongying Zhao[3], Guoqing Chao[4], Yanwei Yu[1,2*]**

[1]Faculty of Information Science and Engineering, Ocean University of China, China
[2]State Key Laboratory of Physical Oceanography, Ocean University of China, China
[3]Shandong University of Science and Technology, China
[4]Department of Software Engineering, Harbin Institute of Technology, China
{liuyuchen1092, kyni}@stu.ouc.edu.cn, zzysuin@163.com, guoqingchao@hit.edu.cn, yanyuwei@ouc.edu.cn

## Abstract

Sequential recommendation models analyze user historical behavior sequences to capture temporal dependencies and the dynamic evolution of interests, enabling accurate predictions of future behaviors. However, there are still two critical challenges that remain unsolved: i) Inadequate temporal modeling of user intent, which fails to distinguish between global intent tendency and temporal contextual intent. ii) Noise in sequential interaction data may introduce bias into the model. To address these issues, we propose a Self-Supervised Hypergraph Sequential Recommendation Framework ($S^2$HyRec). This framework features the Global Intent Tendency module for capturing long-term preferences, the Temporal Contextual Intent module for modeling dynamic time-sensitive interests. Additionally, we develop the Sequence Dependency-Aware module that analyzes the chronological flow of interactions to uncover inherent behavioral dynamics, further enriching the comprehensive user intent representation. To mitigate noisy interactions, we employ a Cross-View Self-Supervised Learning module that enhances the model's ability to distinguish genuine preferences from noise. Extensive experiments on four benchmark datasets demonstrate the superiority of $S^2$HyRec over various state-of-the-art recommendation methods, especially achieving average improvements of 15.13% and 14.03% in NDCG@10 and NDCG@20, respectively, across the four datasets.

**Code** — https://github.com/lyyccccccc/S2HyRec

## Introduction

Personalized recommender systems help users discover items of interest on e-commerce platforms (Ge et al. 2020), video platforms, and social media (Peng et al. 2020). Traditional methods include factorization-based methods and neural collaborative filtering models. Recently, Graph Neural Networks (GNNs) excel at capturing high-order user-item relations. For instance, NGCF (Wang et al. 2019) uses Graph Convolutional Networks (GCNs) for embedding propagation, and GCCF (Chen et al. 2020) demonstrated that nonlinearity is unnecessary in recommenders.

Sequential recommendation analyzes users' temporal interaction patterns to predict future behaviors. As Figure 1 illustrates, user behavior typically exhibits three key elements: (1) global intent tendency, reflecting long-term stable preferences such as consistent purchases of sports-related products; (2) temporal contextual intent, showing time-sensitive interests such as concentrated food purchases within specific timeframes; and (3) noise, including misclicks or sporadic purchases unrelated to general interests. Early works utilized Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and self-attention mechanisms to capture sequential features and temporal dependencies. Further works explored user intent understanding (Li et al. 2023; Chen et al. 2022) by integrating knowledge graphs (KGIN (Wang et al. 2021b)), using disentangled representations (DGCF (Wang et al. 2020b)), and applying contrastive learning (DCCF (Ren et al. 2023)).

While the previously discussed methods have made significant progress in sequential recommendation, two key issues remain less explored: **(1) Inadequate temporal modeling of user intent**: User intent is a dynamic process, influenced by both temporal contextual intent and global intent tendency. Temporal contextual intent represents users' time-sensitive interests and immediate needs, while global intent tendency captures their long-term preferences and habitual behaviors. Existing sequential recommendation models primarily focus on modeling the user's temporal contextual intent but often neglect the global intent tendency across temporal segments. **(2) Noise in Sequential Interaction Data**: In sequential recommendation tasks, noise refers to extraneous data within user behavior sequences that do not reflect genuine user interests or intentions. Such noise may arise from misclicks, sporadic actions, or misleading platform designs. These unpredictable and pattern-deficient behaviors often obscure users' latent preferences, hindering accurate modeling in recommender systems.

To address these challenges, we propose $S^2$HyRec, a Self-Supervised Hypergraph Sequential Recommendation Framework with four core components. To tackle inadequate temporal intent modeling, we develop a global intent tendency learning module employing hypergraph structures to connect users with related items, capturing unified global intent tendency beyond traditional pairwise graphs. We also introduce a temporal contextual intent learning module that leverages GNNs to dynamically model time-sensitive interests across different periods. Additionally, we develop

*Corresponding author.

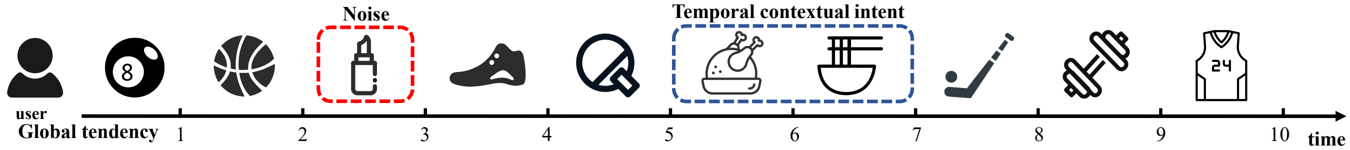



Figure 1: Illustration of the challenges in sequential recommendation

the Sequence Dependency-Aware module that analyzes the chronological flow of interactions to uncover inherent behavioral dynamics, further enriching the comprehensive user intent representation. To address noise in sequential interaction data, we employ a cross-view self-supervised learning module that enhances the robustness of user intent representations by fostering coherence between long-term stable preferences and dynamic time-sensitive interests.

In summary, this paper makes the following contributions:

- We propose a framework that comprehensively models user intent by integrating global tendencies, dynamic temporal contexts, and inherent sequential behaviors.
- We design a cross-view self-supervised learning approach that aligns complementary intent representations, thereby robustly distinguishing genuine preferences from noisy interactions and enhancing preference learning.
- We demonstrate through extensive experiments on four benchmark datasets that $S^2$HyRec outperforms SOTA by 15.13% in NDCG@10 and 14.03% in NDCG@20.

## Related Work

### Sequential Recommendation Systems.

Early sequential recommendation methods, including those based on Markov chains (e.g. Fossil (He and McAuley 2016)), focused on modeling user behavior transitions. The advent of deep learning made neural models (e.g., RNNs (Graves and Graves 2012), CNNs (Xu et al. 2019), and self-attention mechanisms (SASRec (Kang and McAuley 2018), BERT4Rec (Sun et al. 2019))) mainstream, effectively capturing temporal dependencies in interaction sequences. However, these methods primarily model temporal contextual intent but ignore its distinction from global intent tendencies. *Furthermore, their lack of effective noise-filtering can amplify irrelevant patterns, harming accuracy.*

### Recommender Systems with SSL.

Contrastive learning has succeeded across domains (Yuan et al. 2021; Wang et al. 2021a) and shows strong potential in sequential recommendation (Zhou et al. 2021; Zhang et al. 2021; Qin et al. 2023). Methods such as CL4SRec (Xie et al. 2022), DuoRec (Qiu et al. 2022), and ICLRec (Chen et al. 2022) leverage contrastive strategies to uncover latent user intent and improve interest modeling. S3Rec (Zhou et al. 2020) and CoSeRec (Liu et al. 2021) apply multi-scale contrastive learning to capture evolving user interests. Despite these advances, existing SSL-based methods struggle to effectively distinguish noisy from meaningful behavioral patterns. *In contrast, $S^2$HyRec employs a cross-view SSL strategy aligning global and local representations, unlike single-view SSL. This design refines the isolation of genuine user intent from noise and sporadic interactions.*

### Graph-based Recommender Systems.

GNNs have significantly advanced recommendation systems by modeling complex user–item relationships (Huang et al. 2021; Li et al. 2024). Some models, like LightGCN (He et al. 2020) and DGSR (Zhang et al. 2022), simplify or enhance graph propagation. Others, including SURGE (Chang et al. 2021), GCE-GNN (Wang et al. 2020c), and SRGNN (Wu et al. 2019), integrate GNNs with temporal modeling or SSL to better capture dynamic user preferences. SelfGNN (Liu, Xia, and Huang 2024) further constructs SSL tasks directly from graph structures to enhance representation quality. Hypergraph-based approaches effectively capture high-order relationships (Gao et al. 2020; Wang et al. 2020a). MHCN (Yu et al. 2021) employs multi-channel hypergraphs for social recommendations; MBHT (Yang et al. 2022) fuses hypergraph learning with transformers; and SHT (Xia, Huang, and Zhang 2022) combines hypergraph modeling with SSL. *Unlike single-view models that struggle to capture diverse user intent, $S^2$HyRec combines hypergraph-based global modeling with GNN-based temporal learning to represent both shared patterns and personalized dynamics.*

## Preliminaries

**Definition 1 (User and Item Sets)** *Let users be denoted by the set $\mathcal{U} = \{u_1, u_2, \ldots, u_I\}$, and items by $\mathcal{V} = \{v_1, v_2, \ldots, v_J\}$, where I and J represent the total numbers of users and items, respectively.*

**Definition 2 (User Interaction Sequence)** *For each user $u \in \mathcal{U}$, the historical interaction sequence is represented as an ordered list $s_u = [v_1, v_2, \ldots, v_{|s_u|}]$, where each $v_j \in \mathcal{V}$ denotes the $j$-th item interacted with by the user, and $|s_u|$ represents the length of the sequence (i.e., the total number of items the user has interacted with). The sequence is ordered by timestamps $\{t_1, t_2, \ldots, t_{|s_u|}\}$, with $t_1 < t_2 < \cdots < t_{|s_u|}$, where $t_j$ indicates the time at which the user interacted with item $v_j$.*

**Problem 1 (Sequential Recommendation Task)** *The goal of the sequential recommendation task is to predict the next item $v_{|s_u|+1}$ that user $u \in \mathcal{U}$ is most likely to interact with, given their historical interaction sequence $s_u = [v_1, v_2, \ldots, v_{|s_u|}]$. This task can be formulated as a conditional probability maximization problem:*

$$\arg\max_{v_i \in \mathcal{V}} P\left(v_{|s_u|+1} = v_i \mid s_u\right), \tag{1}$$

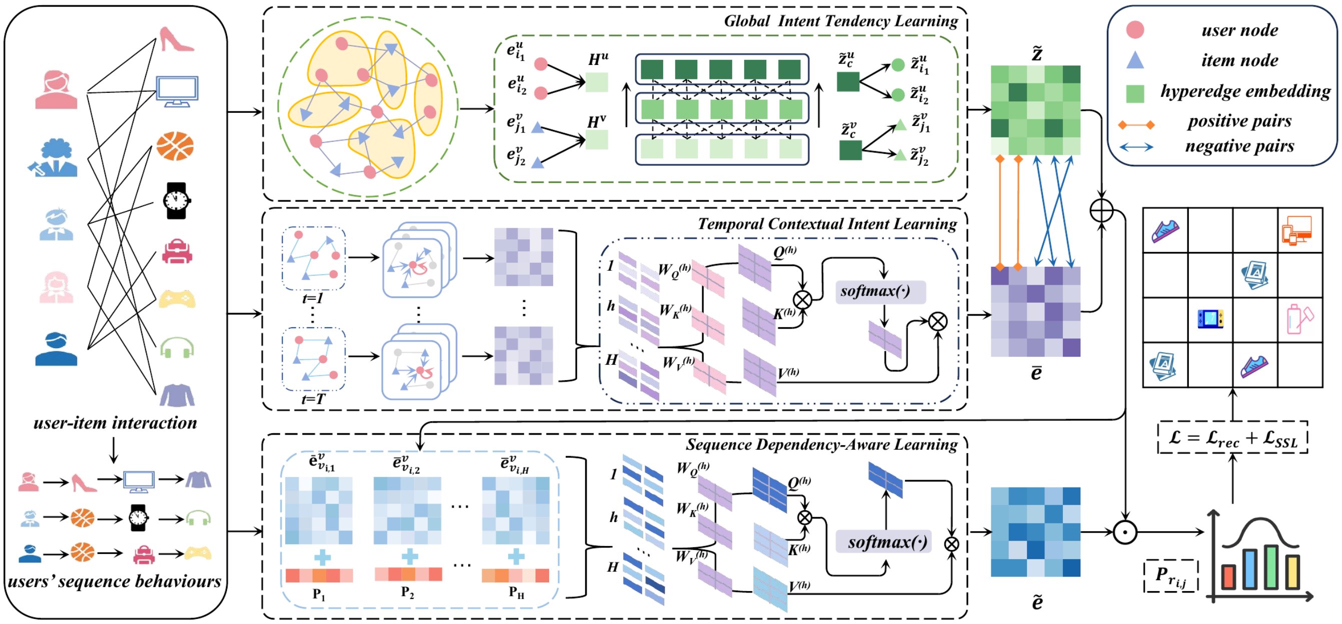



Figure 2: Overall framework of the proposed S$^2$HyRec model.

*where $P\left(v_{|s_u|+1} = v_i \mid s_u\right)$ represents the probability that item $v_i$ will be the next item in the sequence, conditioned on the user's historical interactions $s_u$.*

## Methodology

Figure 2 shows the overall framework of our method. The S$^2$HyRec framework is composed of four core components: *(a) global intent tendency learning*, *(b) temporal contextual intent learning*, *(c)cross-view self-supervised learning* and *(d) sequence dependency-aware learning*.

### Global Intent Tendency Learning

This subsection focuses on modeling the global tendency of user intent. Traditional pairwise interaction graphs are limited in their ability to capture higher-order user-item relationships, often resulting in fragmented representations of user intent (Lizotte, Young, and Allard 2023). *In contrast, hypergraphs can connect multiple users and items simultaneously, thereby enabling more comprehensive modeling of global behavioral patterns.* By unifying related interactions across various time periods, this approach uncovers stable user preferences that are frequently overlooked in analyses constrained to isolated temporal segments. Specifically, we model high-order user-item dependencies globally using learnable hypergraph higher-order dependency matrices $\mathbf{H}^u \in \mathbb{R}^{I \times M}, \mathbf{H}^v \in \mathbb{R}^{J \times M}$:

$$\mathbf{H}^u = \mathbf{E}^u \cdot \mathbf{W}^u, \quad \mathbf{H}^v = \mathbf{E}^v \cdot \mathbf{W}^v, \tag{2}$$

where $\mathbf{W}^u, \mathbf{W}^v \in \mathbb{R}^{d \times M}$ represents the learnable embedding matrices for hyperedges on the user side and the item side, with $d$ as the embedding dimension and $M$ being the number of hyperedges. $\mathbf{E}^u \in \mathbb{R}^{I \times d}, \mathbf{E}^v \in \mathbb{R}^{J \times d}$ represents the embedding matrices of users and items. After learning the hypergraph structure matrices $\mathbf{H}^u$ and $\mathbf{H}^v$, we propagate initial hyperedge embeddings $\mathbf{z}_0^u \in \mathbb{R}^{M \times d}, \mathbf{z}_0^v \in \mathbb{R}^{M \times d}$ through the hypergraph, capturing user-item interactions:

$$\mathbf{z}_k^* = \sigma\left(\mathbf{H}^* \cdot \mathbf{z}_{k-1}^*\right) + \mathbf{z}_{k-1}^*, \text{ where } * \in \{u, v\}. \tag{3}$$

Here, $\sigma(\cdot)$ is a non-linear activation function. For each propagation step $k \in \{1, \ldots, c\}$, this mechanism iteratively refines hyperedge embeddings, effectively capturing high-order user-item dependencies. The parameter $c \in \mathbb{N}^+$ denotes the total number of propagation steps, during which information recursively passes through the hypergraph.

We utilize a hierarchical mapping mechanism to enhance the model's ability to capture complex interactions. This mechanism refines hypergraph embeddings at each step by passing current hyperedge embeddings through fully connected layers with residual connections. Specifically, the hierarchical mapping is expressed as:

$$\tilde{\mathbf{z}}_k^* = \sigma\left(\mathbf{W}_h^* \cdot \mathbf{z}_k^*\right) + \mathbf{z}_k^*, \text{ where } * \in \{u, v\}. \tag{4}$$

where $\mathbf{W}_h^u \in \mathbb{R}^{d \times d}$ is a learnable transformation matrix. This process refines the global intent tendency embeddings through hierarchical feature interactions. Finally, after hypergraph propagation and hierarchical mapping, we obtain the global intent tendency embeddings:

$$\tilde{\mathbf{z}}^u = \sigma\left(\mathbf{H}^u \cdot \tilde{\mathbf{z}}_c^u\right), \quad \tilde{\mathbf{z}}^v = \sigma\left(\mathbf{H}^v \cdot \tilde{\mathbf{z}}_c^v\right). \tag{5}$$

### Temporal Contextual Intent Learning

While hypergraph-based global modeling excels at identifying stable preferences, it may overlook time-sensitive interests that are crucial for accurate recommendations in specific contexts. We employ GNNs to learn temporal contextual intents by slicing sequences into time slices based on timestamps. For each time slice $t$, we define the interaction matrix $\mathbf{A}_t \in \mathbb{R}^{I\times J}$ to represent implicit user-item relationships. $A_{t,i,j}$ is 1 if user $u_i$ interacted with item $v_j$ during the $t$-th interval, 0 otherwise. We project each user and item into a $d$-dimensional latent space, forming embedding matrices $\mathbf{E}^u_t \in \mathbb{R}^{I\times d}$ and $\mathbf{E}^v_t \in \mathbb{R}^{J\times d}$ for the $t$-th period. For the $l_g$-th GNN layer, message-passing is:

$$\begin{aligned} \mathbf{e}^u_{t,i,l_g} &= \sigma(\mathbf{A}_{t,i,*} \cdot \mathbf{E}^v_{t,l_g-1}), \\ \mathbf{e}^v_{t,j,l_g} &= \sigma(\mathbf{A}^\top_{t,*,j} \cdot \mathbf{E}^u_{t,l_g-1}), \end{aligned} \tag{6}$$

where $\mathbf{e}^u_{t,i,l_g}, \mathbf{e}^v_{t,j,l_g} \in \mathbb{R}^d$ denote information aggregated from neighboring nodes to $u_i$ and $v_j$, and $\sigma(\cdot)$ is the activation function (e.g., LeakyReLU). We adopt random edge dropout to mitigate overfitting. We average all GNN layers to obtain temporal contextual intent embeddings:

$$\mathbf{e}^u_{t,i} = \frac{1}{L_g}\sum_{l_g=1}^{L_g} \mathbf{e}^u_{t,i,l_g}, \quad \mathbf{e}^v_{t,j} = \frac{1}{L_g}\sum_{l_g=1}^{L_g} \mathbf{e}^v_{t,j,l_g}, \tag{7}$$

where $L_g \in \mathbb{N}^+$ is the total number of GNN layers. Next, we build temporal embedding sequences for users and items:

$$\begin{aligned} \mathbf{S}^{temp}_i &= \left[\mathbf{e}^u_{1,i}; \mathbf{e}^u_{2,i}; \ldots; \mathbf{e}^u_{T,i}\right]^\top, \\ \mathbf{S}^{temp}_j &= \left[\mathbf{e}^v_{1,j}; \mathbf{e}^v_{2,j}; \ldots; \mathbf{e}^v_{T,j}\right]^\top, \end{aligned} \tag{8}$$

where $\mathbf{e}^u_{t,i}, \mathbf{e}^v_{t,j} \in \mathbb{R}^d$ are the temporal contextual intent embeddings of user $u_i$ and item $v_j$ at the $t$-th time interval, respectively. To capture dependencies across time intervals, we employ a multi-head dot-product attention mechanism:

$$\mathbf{E}^u_i = \text{Self-Att}(\mathbf{S}^{temp}_i), \quad \mathbf{E}^v_j = \text{Self-Att}(\mathbf{S}^{temp}_j), \tag{9}$$

where Self-Att$(\cdot)$ denotes the self-attention operation. The output $\mathbf{E}^u_i, \mathbf{E}^v_j \in \mathbb{R}^{T\times d}$ are the refined, contextually enriched temporal contextual intent embedding matrices. The final temporal contextual intent representations $\bar{\mathbf{e}}^u_i$ and $\bar{\mathbf{e}}^v_j$ are obtained by summing embeddings across time intervals:

$$\bar{\mathbf{e}}^u_i = \sum_{t=1}^{T} \mathbf{E}^u_{i,t}, \quad \bar{\mathbf{e}}^v_j = \sum_{t=1}^{T} \mathbf{E}^v_{j,t}, \tag{10}$$

where $\mathbf{E}^u_{i,t} \in \mathbb{R}^d$ and $\mathbf{E}^v_{j,t} \in \mathbb{R}^d$ are attention-enhanced embeddings at time interval $t$, with cross-temporal context.

## Cross-View Self-Supervised Learning

To simultaneously address noise in sequential interaction data and further strengthen the modeling of long-term user intent, we propose a Cross-View Self-Supervised Learning module. This module enhances the robustness of user intent representations by explicitly promoting coherence between users' long-term stable preferences and their dynamic, time-sensitive interests. While genuine user preferences naturally maintain consistency across these fundamental yet complementary views, noisy interactions often introduce discrepancies and disrupt this alignment, especially at different temporal scales. By enforcing consistency through contrastive learning, our module not only filters out noise-induced divergences, but also facilitates more effective and in-depth modeling of users' persistent intent. Specifically, we introduce a projection matrix $\mathbf{W} \in \mathbb{R}^{d\times d}$ to map temporal contextual intent embeddings to align with global intent tendency embeddings. The self-supervised learning (SSL) loss is:

$$\mathcal{L}_{SSL} = \mathcal{L}_{\text{user}} + \mathcal{L}_{\text{item}}. \tag{11}$$

The user contrastive loss $\mathcal{L}_{\text{user}}$ is defined as:

$$\mathcal{L}_{\text{user}} = -\sum_{u_i\in\mathcal{U}} \log \frac{\exp(\text{sim}(\mathbf{W}\bar{\mathbf{e}}^u_i, \tilde{\mathbf{z}}^u_i)/\tau)}{\sum_{u_{i'}\in\mathcal{U}} \exp(\text{sim}(\mathbf{W}\bar{\mathbf{e}}^u_i, \tilde{\mathbf{z}}^u_{i'})/\tau)}, \tag{12}$$

where $\bar{\mathbf{e}}^u_i$ and $\tilde{\mathbf{z}}^u_i$ denote temporal contextual intent and global intent tendency embeddings for user $u_i$, respectively. sim$(\cdot)$ represents the cosine similarity, and $\tau \in \mathbb{R}^+$ is the temperature parameter adjusting the softmax scale. The item contrastive loss $\mathcal{L}_{\text{item}}$ is computed similarly.

## Sequence Dependency-Aware Learning

Complementary to the global and temporal contextual intent modeling, we develop the Sequence Dependency-Aware module that analyzes the chronological flow of interactions to uncover inherent behavioral dynamics, thereby enriching the comprehensive user intent representation. For user $u_i$, the interaction sequence is:

$$\mathbf{S}^{att}_{i,0} = \left(\bar{\mathbf{e}}^v_{v_{i,1}} + \mathbf{p}_1, \bar{\mathbf{e}}^v_{v_{i,2}} + \mathbf{p}_2, \ldots, \bar{\mathbf{e}}^v_{v_{i,N}} + \mathbf{p}_N\right), \tag{13}$$

where $\bar{\mathbf{e}}^v_{v_{i,n}} \in \mathbb{R}^d$ is the embedding of $n$-th item $v_{i,n}$ interacted by user $u_i$, $\mathbf{p}_n \in \mathbb{R}^d$ is the positional embedding encoding item position, and $N$ denotes sequence length. Contextual representations are computed via multi-head self-attention, with the sequence representation updated at each layer $l_a \in \{1, \ldots, L_a\}$ as:

$$\mathbf{S}^{att}_{i,l_a} = \text{LayerNorm}\left(\mathbf{S}^{att}_{i,l_a-1} + \text{Self-Att}(\mathbf{S}^{att}_{i,l_a-1})\right), \tag{14}$$

where Self-Att$(\mathbf{S}^{att}_{i,l_a-1})$ aggregates dependencies between different positions in the sequence. The final representation for user $u_i$ is obtained by aggregating the layer outputs:

$$\tilde{\mathbf{e}}^u_i = \sum_{n=1}^{N} \mathbf{S}^{att}_{i,L_a,n}, \tag{15}$$

where $\mathbf{S}^{att}_{i,L_a,n}$ denotes the embedding at the $n$-th position in the final layer output. The resulting $\tilde{\mathbf{e}}^u_i \in \mathbb{R}^d$ captures the temporal dependency in user $u_i$'s interaction history.

## Dual-Task Training

Integrating multiple user intent views is crucial for comprehensive modeling. We propose dual-task training that balances these complementary perspectives to create a robust recommendation model resilient to noise and capable of capturing long-term preferences and contextual interests. We integrate global intent tendency embeddings ($\tilde{\mathbf{z}}^u_i$ and $\tilde{\mathbf{z}}^v_j$) and temporal contextual intent embeddings ($\bar{\mathbf{e}}^u_i$ and $\bar{\mathbf{e}}^v_j$) using hyperparameter $\alpha$, which controls their relative weight:

$$\hat{\mathbf{e}}^u_i = (1-\alpha)\cdot\bar{\mathbf{e}}^u_i + \alpha\cdot\tilde{\mathbf{z}}^u_i, \quad \hat{\mathbf{e}}^v_j = (1-\alpha)\cdot\bar{\mathbf{e}}^v_j + \alpha\cdot\tilde{\mathbf{z}}^v_j. \tag{16}$$

The predicted score for user $u_i$ interacting with item $v_j$ is then computed as:

$$Pr_{i,j} = \hat{\mathbf{e}}^{u\top}_i \cdot \hat{\mathbf{e}}^v_j + \tilde{\mathbf{e}}^{u\top}_i \cdot \hat{\mathbf{e}}^v_j, \tag{17}$$

where $\tilde{\mathbf{e}}_i^u$ represents the sequential dependency embeddings for user $u_i$. This predicted score integrates both the user's personalized preferences and collective behavior patterns, capturing multiple dimensions of user behavior, including global intent tendency, temporal contextual intent, and sequential dependencies. Next, positive samples are user-interacted items, while negative samples are non-interacted ones. To prevent the predicted values from becoming arbitrarily large, we optimize the following loss function:

$$\mathcal{L}_{rec} = \sum_{i=1}^{I} \sum_{k=1}^{N_{pr}} \max\left(0, 1 - Pr_{i,p_k} + Pr_{i,n_k}\right), \qquad (18)$$

where $N_{pr} \in \mathbb{N}^+$ denotes the number of samples, and $p_k$ and $n_k$ denote the indices of the $k$-th positive and negative items, respectively. To prevent overfitting and stabilize training, the following loss function is optimized:

$$\mathcal{L} = \mathcal{L}_{rec} + \epsilon_1 \cdot \mathcal{L}_{SSL} + \epsilon_2 \cdot \|\mathbf{\Theta}\|_F^2, \qquad (19)$$

where $\mathcal{L}_{rec}$ is the prediction loss, $\mathcal{L}_{SSL}$ is the cross-view self-supervised learning loss, and $\|\mathbf{\Theta}\|_F^2$ is the weight decay regularization. Coefficients $\epsilon_1, \epsilon_2 \in \mathbb{R}^+$ balance the respective term contributions.

# Experiments

## Datasets and Baselines

The experiments are conducted on four open-source datasets, as summarized in Table 1. To evaluate our proposed S$^2$HyRec, we compare it with eighteen state-of-the-art baselines from different research areas, including collaborative filtering methods (BiasMF (Koren, Bell, and Volinsky 2009) and NCF (He et al. 2017)), traditional sequential recommendation approaches (GRU4Rec (Hidasi 2015), SASRec (Kang and McAuley 2018), TiSASRec (Li, Wang, and McAuley 2020) and Bert4Rec (Sun et al. 2019)), GNN-based methods (NGCF (Wang et al. 2019), LightGCN (He et al. 2020), SRGNN (Wu et al. 2019), GCE-GNN (Wang et al. 2020c), SURGE (Chang et al. 2021) and DGCF (Li et al. 2020)), and self-supervised learning methods (SGL (Wu et al. 2021), ICLRec (Chen et al. 2022), CoSeRec (Liu et al. 2021), CoTRec (Xia et al. 2021), CLSR (Zheng et al. 2022) and SelfGNN (Liu, Xia, and Huang 2024)).

| **Dataset** | **#User** | **#Item** | **#Interaction** | **Density** |
|---|---|---|---|---|
| Gowalla | 48,653 | 52,621 | 1,807,125 | $7.1 \times 10^{-4}$ |
| Movielens | 24,312 | 8,688 | 1,758,929 | $8.3 \times 10^{-3}$ |
| Yelp | 19,751 | 38,391 | 1,467,157 | $1.9 \times 10^{-3}$ |
| Tmall | 212,077 | 99,193 | 3,938,991 | $1.9 \times 10^{-4}$ |

Table 1: Statistics of the experimental datasets.

## Experiment Settings and Evaluation Settings

S$^2$HyRec is implemented with TensorFlow using Adam optimizer (learning rate: $1 \times 10^{-3}$, decay ratio: 0.96) and embedding dimension of 64. We tune key hyperparameters including the number of graph neural layers {1-5}, batch size $\{128, 256, 512\}$, attention layers {1-5}, fusion weight $\alpha$ {0.1-0.6}, SSL loss weight $\epsilon_1$ {$10^{-3}$-$10^{-7}$}, and regularization weight $\epsilon_2$ {$10^{-2}$-$10^{-5}$}. The dropout rate is set to 0.5. All experiments are conducted on an NVIDIA GeForce RTX 4090 24GB GPU. Performance is evaluated using Hit Rate (HR@K) and Normalized Discounted Cumulative Gain (NDCG@K), abbreviated as H@K and N@K.

## Performance Comparison

As shown in Table 3 and Table 2, S$^2$HyRec consistently outperforms all baselines across all datasets. This superior performance is attributed to the synergistic effect of its four core components. Notably, S$^2$HyRec achieves substantial improvements, with up to 15.1% gain in NDCG@10 on Gowalla and 18.1% in NDCG@20 on MovieLens. The performance advantage extends to the larger-scale Tmall dataset, where S$^2$HyRec achieves 10.1% improvement in HR@10 and 16.1% in NDCG@10 compared to SelfGNN, demonstrating strong generalization across diverse domains.

S$^2$HyRec's dual-perspective intent learning effectively balances global and temporal contextual modeling. Unlike traditional GNNs (NGCF, LightGCN), our hypergraph approach transcends bipartite limitations, enabling multi-user-item connections for complex relationships. The temporal contextual intent learning module overcomes sequential method limitations (SASRec, Bert4Rec) via effective time slicing and attention mechanisms. Our cross-view self-supervised learning enhances noise robustness by aligning global and temporal perspectives. Unlike existing methods (SGL, ICLRec), we directly optimize for cross-perspective consistency, leveraging the principle that genuine preferences remain consistent across views while noise creates discrepancies. The sequence dependency component captures sequential dependencies in user interaction history, significantly improving performance on complex datasets.

| Metrics | SASRec | LightGCN | SRGNN | SelfGNN | **Ours** | **Impro(%)** |
|---|---|---|---|---|---|---|
| H@10 | 0.214 | 0.219 | 0.235 | 0.257 | **0.283** | **10.12%** |
| H@20 | 0.283 | 0.296 | 0.320 | 0.344 | **0.368** | **6.98%** |
| N@10 | 0.145 | 0.145 | 0.152 | 0.155 | **0.180** | **16.13%** |
| N@20 | 0.162 | 0.164 | 0.173 | 0.177 | **0.202** | **14.12%** |

Table 2: Performance comparison on Tmall dataset.

## Ablation Study

As shown in Figure 3, we assess each S$^2$HyRec module via six variants:: (1) *w/o gta*: Removes the global intent tendency learning component; (2) *w/o hg*: Replaces the hypergraph with a standard bipartite graph in global intent tendency learning; (3) *w/o att*: Replaces attention with simple summation for temporal contextual intent learning; (4) *w/o gru-att*: Uses GRU instead of attention for temporal contextual intent learning; (5) *w/o ssl*: Disables contrastive learning between global and temporal representations; (6) *w/o seq*: Removes the sequence dependency-aware component.

Consistent performance degradation of **w/o gta** and **w/o hg** variants across all datasets, especially on Yelp, highlights

| Methods | Gowalla | | | | MovieLens | | | | Yelp | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | H@10 | H@20 | N@10 | N@20 | H@10 | H@20 | N@10 | N@20 | H@10 | H@20 | N@10 | N@20 |
| BiasMF | 0.543 | 0.669 | 0.362 | 0.393 | 0.208 | 0.311 | 0.115 | 0.141 | 0.281 | 0.411 | 0.151 | 0.184 |
| NCF | 0.606 | 0.716 | 0.418 | 0.446 | 0.198 | 0.299 | 0.107 | 0.133 | 0.313 | 0.462 | 0.169 | 0.206 |
| GRU4Rec | 0.393 | 0.515 | 0.253 | 0.284 | 0.127 | 0.238 | 0.076 | 0.099 | 0.087 | 0.143 | 0.043 | 0.057 |
| SASRec | 0.562 | 0.688 | 0.360 | 0.392 | 0.121 | 0.163 | 0.072 | 0.082 | 0.106 | 0.151 | 0.057 | 0.057 |
| TiSASRec | 0.573 | 0.690 | 0.376 | 0.405 | 0.120 | 0.165 | 0.071 | 0.082 | 0.092 | 0.133 | 0.051 | 0.061 |
| Bert4Rec | 0.544 | 0.676 | 0.359 | 0.393 | 0.175 | 0.299 | 0.087 | 0.118 | 0.290 | 0.428 | 0.158 | 0.193 |
| NGCF | 0.550 | 0.692 | 0.347 | 0.383 | 0.147 | 0.238 | 0.076 | 0.099 | 0.325 | 0.475 | 0.174 | 0.212 |
| LightGCN | 0.456 | 0.592 | 0.286 | 0.318 | 0.211 | 0.319 | 0.115 | 0.142 | 0.346 | 0.495 | 0.189 | 0.226 |
| SRGNN | 0.575 | 0.687 | 0.371 | 0.399 | 0.122 | 0.166 | 0.069 | 0.080 | 0.097 | 0.137 | 0.054 | 0.064 |
| GCE-GNN | 0.578 | 0.709 | 0.373 | 0.406 | 0.169 | 0.264 | 0.088 | 0.112 | 0.297 | 0.440 | 0.157 | 0.194 |
| SURGE | 0.479 | 0.654 | 0.259 | 0.304 | 0.237 | 0.337 | 0.124 | 0.149 | 0.278 | 0.428 | 0.144 | 0.182 |
| DGCF | 0.524 | 0.629 | 0.339 | 0.388 | 0.216 | 0.309 | 0.116 | 0.136 | 0.318 | 0.452 | 0.168 | 0.202 |
| SGL | 0.413 | 0.495 | 0.259 | 0.280 | 0.223 | 0.282 | 0.120 | 0.135 | 0.320 | 0.461 | 0.170 | 0.206 |
| ICLRec | 0.197 | 0.265 | 0.122 | 0.193 | 0.190 | 0.248 | 0.126 | 0.140 | 0.195 | 0.274 | 0.113 | 0.133 |
| CoSeRec | 0.556 | 0.669 | 0.362 | 0.390 | 0.166 | 0.221 | 0.112 | 0.125 | 0.195 | 0.270 | 0.110 | 0.129 |
| CoTRec | 0.531 | 0.608 | 0.400 | 0.419 | 0.216 | 0.311 | 0.117 | 0.141 | 0.283 | 0.416 | 0.152 | 0.185 |
| CLSR | 0.529 | 0.699 | 0.296 | 0.339 | 0.061 | 0.076 | 0.036 | 0.040 | 0.261 | 0.400 | 0.134 | 0.169 |
| SelfGNN | 0.637 | 0.745 | 0.437 | 0.465 | 0.216 | 0.328 | 0.117 | 0.146 | 0.348 | 0.496 | 0.190 | 0.227 |
| **S²HyRec** | **0.686** | **0.781** | **0.503** | **0.527** | **0.269** | **0.384** | **0.147** | **0.176** | **0.390** | **0.539** | **0.214** | **0.251** |
| **Improve (%)** | **7.7%** | **4.8%** | **15.1%** | **13.3%** | **13.5%** | **13.9%** | **16.7%** | **18.1%** | **12%** | **8.7%** | **12.6%** | **10.6%** |

Table 3: Performance comparison on Gowalla, MovieLens, and Yelp datasets in terms of HR@K and NDCG@K. The improvement over the best-performed baseline method is statistically significant with $p < 0.01$.

the global intent tendency learning module's necessity for capturing unified interactions and stable patterns, and hypergraphs' superiority in modeling richer, higher-order dependencies. Significant performance degradation in **w/o att** and **w/o gru-att** variants shows attention better captures cross-interval dependencies than simple aggregation (summation or GRU), highlighting the temporal contextual intent learning module's crucial role in integrating users' dynamic, time-sensitive interests. The **w/o ssl** variant's drop indicates that cross-view self-supervision is vital for noise robustness by aligning authentic user preferences between global and temporal views. Likewise, the **w/o seq** variant's decline shows that the sequence dependency-aware module is key to parsing chronological interactions to expose behavioral dynamics and enrich user-intent representations.

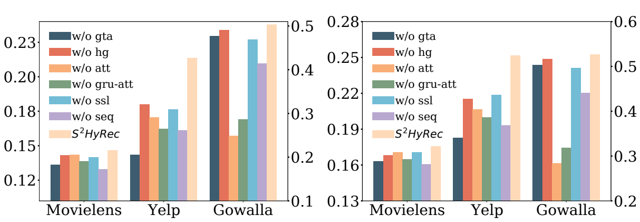
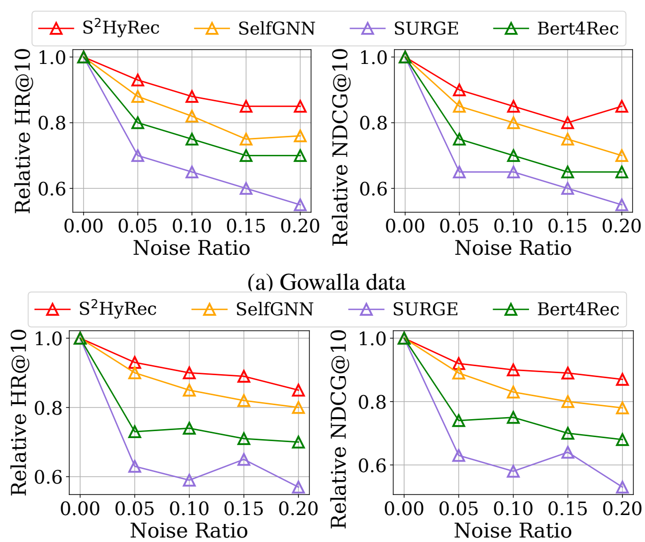



Figure 3: Evaluation results for ablation study in NDCG@10 (left) and NDCG@20 (right), MovieLens and Yelp (left y-axis), and Gowalla (right y-axis).

## Noise Robustness Study

To evaluate model robustness against noise, as shown in Figure 4, we randomly replaced 5%-20% of real interactions




(a) Gowalla data

(b) Movielens data

Figure 4: Performance resilience against noise corruption.

with fake items in user sequences, comparing S²HyRec, SelfGNN, SURGE, and Bert4Rec. S²HyRec demonstrates superior noise resistance across all datasets. At 20% noise, it maintains approximately 85% of original HR@10 and 82% of NDCG@10 on Gowalla, and about 85% of HR@10 and 87% of NDCG@10 on Movielens. In contrast, baselines like SURGE significantly deteriorate, dropping to 55%-60% performance at 20% noise, while SelfGNN maintains 70%-80% and Bert4Rec shows moderate robustness. This supe-

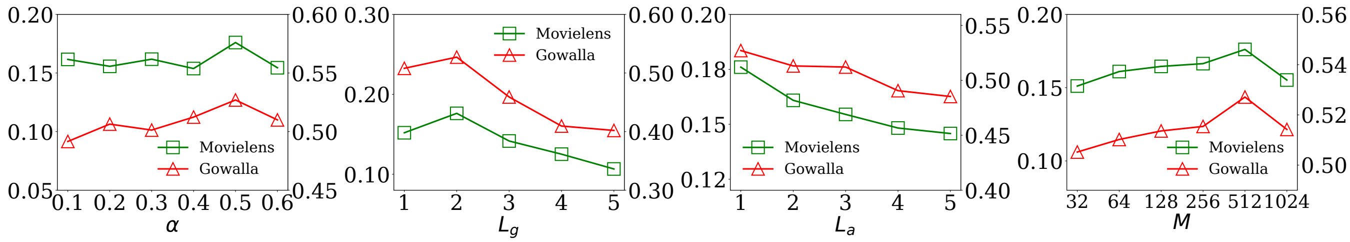



Figure 5: Evaluation results *w.r.t.* $\alpha$, $L_g$, $L_a$, and $M$ in NDCG@20, MovieLens on left y-axis and Gowalla on right y-axis.

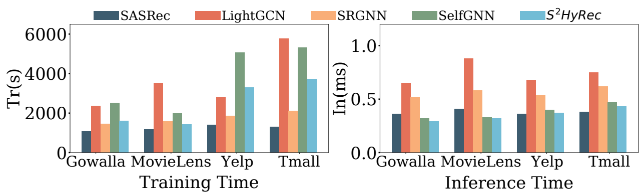



Figure 6: Training and Inference Efficiency.

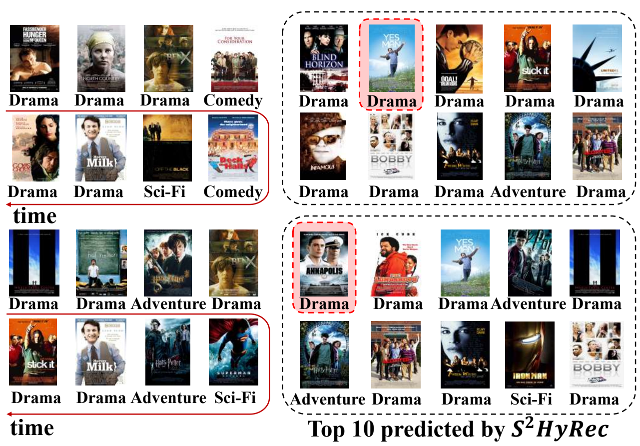



Figure 7: Case study

riority stems from S²HyRec's dual modeling approach and cross-view self-supervised learning, which effectively aligns global and temporal contextual intents, enabling it to distinguish genuine user interests from noise and filter interference signals. This makes S²HyRec highly suitable for real-world scenarios where noise is inevitable.

## Hyperparameters Analysis

We study how S²HyRec's key hyperparameters influence performance (Figure 5). The fusion weight $\alpha$, which balances global and temporal intent, peaks at 0.5 for MovieLens and Gowalla; values above 0.6 degrade accuracy as temporal context fades. For the number of GNN layers $L_g$, experiments show that $L_g = 2$ provides optimal depth for effective information aggregation, while deeper stacks overfit sparse data. The attention layers $L_a$ favor a single layer, suggesting that a single attention layer sufficiently captures essential temporal dependencies across our datasets. Lastly, we examine the number of hyperedges $M$, and that $M = 512$ achieves the best performance by enabling rich high-order relationship modeling while maintaining computational efficiency.

## Efficiency Comparison

As shown in Figure 6, S²HyRec demonstrates competitive computational efficiency. For training time, it is significantly faster than LightGCN (31%-35% reduction) and SelfGNN (29%-35% reduction) across all datasets. Compared to SASRec and SRGNN, S²HyRec shows moderate increases due to its multi-component architecture, justified by substantial performance gains. For inference, S²HyRec achieves the lowest latency (0.29-0.43ms) among all models, making it suitable for real-time recommendations. Efficiency stems from parallel computation and efficient hypergraph design.

## Case Study

This section illustrates S²HyRec's practical application, using two user examples to demonstrate its capability in predicting users' next movies based on historical behavior. For User 6582 (upper part of Figure 7), a dominant global intent for *Drama* emerges, with brief temporal shifts to *Comedy* and occasional *Sci-Fi* noise; the model's top-10 predictions stay mostly within *Drama* and correctly include the ground truth "*Yes Man (2006)*", while other genres appear sparsely. Similarly, for User 5923 (lower part of Figure 7), a strong global *Drama* intent is exhibited, alongside temporal shifts to *Adventure* and *Sci-Fi* as noise. The model's predictions reflect this dominant *Drama* preference, correctly identifying the ground truth ("*Annapolis (2006)*"), and ranking other genres lower. This case study highlights S²HyRec's effectiveness in leveraging historical user behavior to capture both *temporal intent* and *global trends*, thereby providing accurate dynamic interest prediction and personalized recommendations in real-world applications.

# Conclusion

In this paper, we propose S²HyRec, a self-supervised hypergraph framework for sequential recommendation. Our framework addresses two key challenges: inadequate temporal modeling of user intent and noise in interaction sequences. S²HyRec effectively captures both global intent tendencies and temporal contextual intent while filtering noisy interactions. Experiments on four benchmark datasets demonstrate S²HyRec outperforms state-of-the-art methods. Future work will explore adaptive hypergraph structures that evolve with user preferences.

## Acknowledgments

This work is partially supported by the National Natural Science Foundation of China under grant Nos. 62176243, 62472263, and 62276079, and the Fundamental Research Funds for the Central Universities under grant No. 202442005.

## References

Chang, J.; Gao, C.; Zheng, Y.; Hui, Y.; Niu, Y.; Song, Y.; Jin, D.; and Li, Y. 2021. Sequential recommendation with graph neural networks. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, 378–387.

Chen, L.; Wu, L.; Hong, R.; Zhang, K.; and Wang, M. 2020. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 27–34.

Chen, Y.; Liu, Z.; Li, J.; McAuley, J.; and Xiong, C. 2022. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*, 2172–2182.

Gao, Y.; Zhang, Z.; Lin, H.; Zhao, X.; Du, S.; and Zou, C. 2020. Hypergraph learning: Methods and practices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5): 2548–2566.

Ge, Y.; Zhao, S.; Zhou, H.; Pei, C.; Sun, F.; Ou, W.; and Zhang, Y. 2020. Understanding echo chambers in e-commerce recommender systems. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 2261–2270.

Graves, A.; and Graves, A. 2012. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, 37–45.

He, R.; and McAuley, J. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th international conference on data mining (ICDM)*, 191–200. IEEE.

He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 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.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, 173–182.

Hidasi, B. 2015. Session-based Recommendations with Recurrent Neural Networks. *arXiv preprint arXiv:1511.06939*.

Huang, C.; Chen, J.; Xia, L.; Xu, Y.; Dai, P.; Chen, Y.; Bo, L.; Zhao, J.; and Huang, J. X. 2021. Graph-enhanced multi-task learning of multi-level transition dynamics for session-based recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 4123–4130.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, 197–206. IEEE.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37.

Li, J.; Wang, Y.; and McAuley, J. 2020. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th international conference on web search and data mining*, 322–330.

Li, S.; Chen, W.; Wang, B.; Huang, C.; Yu, Y.; and Dong, J. 2024. Mcn4rec: Multi-level collaborative neural network for next location recommendation. *ACM Transactions on Information Systems*, 42(4): 1–26.

Li, X.; Sun, A.; Zhao, M.; Yu, J.; Zhu, K.; Jin, D.; Yu, M.; and Yu, R. 2023. Multi-intention oriented contrastive learning for sequential recommendation. In *Proceedings of the sixteenth ACM international conference on web search and data mining*, 411–419.

Li, X.; Zhang, M.; Wu, S.; Liu, Z.; Wang, L.; and Philip, S. Y. 2020. Dynamic graph collaborative filtering. In *2020 IEEE international conference on data mining (ICDM)*, 322–331. IEEE.

Liu, Y.; Xia, L.; and Huang, C. 2024. Selfgnn: Self-supervised graph neural networks for sequential recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1609–1618.

Liu, Z.; Chen, Y.; Li, J.; Yu, P. S.; McAuley, J.; and Xiong, C. 2021. Contrastive self-supervised sequential recommendation with robust augmentation. *arXiv preprint arXiv:2108.06479*.

Lizotte, S.; Young, J.-G.; and Allard, A. 2023. Hypergraph reconstruction from uncertain pairwise observations. *Scientific Reports*, 13(1): 21364.

Peng, Z.; Huang, W.; Luo, M.; Zheng, Q.; Rong, Y.; Xu, T.; and Huang, J. 2020. Graph representation learning via graphical mutual information maximization. In *Proceedings of The Web Conference 2020*, 259–270.

Qin, X.; Yuan, H.; Zhao, P.; Fang, J.; Zhuang, F.; Liu, G.; Liu, Y.; and Sheng, V. 2023. Meta-optimized contrastive learning for sequential recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 89–98.

Qiu, R.; Huang, Z.; Yin, H.; and Wang, Z. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, 813–823.

Ren, X.; Xia, L.; Zhao, J.; Yin, D.; and Huang, C. 2023. Disentangled contrastive collaborative filtering. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1137–1146.

Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; and Jiang, P. 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.

Wang, J.; Ding, K.; Hong, L.; Liu, H.; and Caverlee, J. 2020a. Next-item recommendation with sequential hypergraphs. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 1101–1110.

Wang, J.; Yuan, F.; Chen, J.; Wu, Q.; Yang, M.; Sun, Y.; and Zhang, G. 2021a. Stackrec: Efficient training of very deep sequential recommender models by iterative stacking. In *Proceedings of the 44th International ACM SIGIR conference on Research and Development in Information Retrieval*, 357–366.

Wang, X.; He, X.; Wang, M.; Feng, F.; and Chua, T.-S. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 165–174.

Wang, X.; Huang, T.; Wang, D.; Yuan, Y.; Liu, Z.; He, X.; and Chua, T.-S. 2021b. Learning intents behind interactions with knowledge graph for recommendation. In *Proceedings of the web conference 2021*, 878–887.

Wang, X.; Jin, H.; Zhang, A.; He, X.; Xu, T.; and Chua, T.-S. 2020b. Disentangled graph collaborative filtering. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 1001–1010.

Wang, Z.; Wei, W.; Cong, G.; Li, X.-L.; Mao, X.-L.; and Qiu, M. 2020c. Global context enhanced graph neural networks for session-based recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 169–178.

Wu, J.; Wang, X.; Feng, F.; He, X.; Chen, L.; Lian, J.; and Xie, X. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, 726–735.

Wu, S.; Tang, Y.; Zhu, Y.; Wang, L.; Xie, X.; and Tan, T. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 346–353.

Xia, L.; Huang, C.; and Zhang, C. 2022. Self-supervised hypergraph transformer for recommender systems. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 2100–2109.

Xia, X.; Yin, H.; Yu, J.; Shao, Y.; and Cui, L. 2021. Self-supervised graph co-training for session-based recommendation. In *Proceedings of the 30th ACM international conference on information & knowledge management*, 2180–2190.

Xie, X.; Sun, F.; Liu, Z.; Wu, S.; Gao, J.; Zhang, J.; Ding, B.; and Cui, B. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*, 1259–1273. IEEE.

Xu, C.; Zhao, P.; Liu, Y.; Xu, J.; S. Sheng, V. S. S.; Cui, Z.; Zhou, X.; and Xiong, H. 2019. Recurrent convolutional neural network for sequential recommendation. In *The world wide web conference*, 3398–3404.

Yang, Y.; Huang, C.; Xia, L.; Liang, Y.; Yu, Y.; and Li, C. 2022. Multi-behavior hypergraph-enhanced transformer for sequential recommendation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 2263–2274.

Yu, J.; Yin, H.; Li, J.; Wang, Q.; Hung, N. Q. V.; and Zhang, X. 2021. Self-supervised multi-channel hypergraph convolutional network for social recommendation. In *Proceedings of the web conference 2021*, 413–424.

Yuan, X.; Duan, D.; Tong, L.; Shi, L.; and Zhang, C. 2021. Icai-sr: Item categorical attribute integrated sequential recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, 1687–1691.

Zhang, M.; Wu, S.; Yu, X.; Liu, Q.; and Wang, L. 2022. Dynamic graph neural networks for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(5): 4741–4753.

Zhang, S.; Yao, D.; Zhao, Z.; Chua, T.-S.; and Wu, F. 2021. Causerec: Counterfactual user sequence synthesis for sequential recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 367–377.

Zheng, Y.; Gao, C.; Chang, J.; Niu, Y.; Song, Y.; Jin, D.; and Li, Y. 2022. Disentangling long and short-term interests for recommendation. In *Proceedings of the ACM Web Conference 2022*, 2256–2267.

Zhou, C.; Ma, J.; Zhang, J.; Zhou, J.; and Yang, H. 2021. Contrastive learning for debiased candidate generation in large-scale recommender systems. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 3985–3995.

Zhou, K.; Wang, H.; Zhao, W. X.; Zhu, Y.; Wang, S.; Zhang, F.; Wang, Z.; and Wen, J.-R. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*, 1893–1902.