Intent Oriented Contrastive Learning for Sequential Recommendation

Wuhong Wang, Jianhui Ma*, Yuren Zhang, Kai Zhang, Junzhe Jiang, Yihui Yang, Yacong Zhou, Zheng Zhang

State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China, China
wangwuhong@mail.ustc.edu.cn, jianhui@ustc.edu.cn

Abstract

Sequential recommendation aims to predict the next item a user is likely to interact with based on their historical interaction sequence. Capturing user intent is crucial in this process, as each interaction is typically driven by specific intentions (e.g., buying skincare products for skin maintenance, buying makeup for cosmetic purposes, etc.). However, users often have multiple, dynamically changing intents, making it challenging for models to accurately learn these intents when relying on the entire historical sequence as input. To address this, we propose a novel framework called Intent Oriented Contrastive Learning for Sequential Recommendation (IOCLRec). This framework begins by segmenting users' sequential behaviors into multiple subsequences, which represent the coarse-grained intents of users at different points in their interaction history. These subsequences form the basis for the three contrastive learning modules within IOCLRec. The fine-grained intent contrastive learning module uncovers detailed intent representations, while the single-intent and multi-intent contrastive learning modules utilize intent-oriented data augmentation operators to capture the diverse intents of users. These three modules work synergistically, driving comprehensive performance optimization in intricate sequential recommendation scenarios. Our method has been extensively evaluated on four public datasets, demonstrating superior effectiveness.

1 Introduction

Recommendation systems have become indispensable tools for helping users navigate the vast array of available products and services (Zhang et al. 2021b). Sequential Recommendation (SR), a specialized field within recommendation systems, aims to uncover temporal patterns in user interactions to predict future behaviors (Roy and Dutta 2022). The core approach to addressing SR challenges involves modeling the dynamic nature of user preferences by analyzing sequences of their past interactions (Qiu, Huang, and Yin 2021; Dang et al. 2023).

Users' interactions with items are driven by specific intents (e.g., purchasing running shoes for exercise, buying books for reading, etc.). Accurately identifying user intents is crucial for understanding user behavior and has significant potential to enhance and interpret recommendation systems (Chen et al. 2022; Zhang et al. 2021a). Existing research primarily focuses on modeling user intent in the latent space (Ma et al. 2020; Wu et al. 2023; Wang et al. 2024). Recently, some studies have incorporated Self-Supervised Learning (SSL) into intent modeling (Chen et al. 2022; Li et al. 2023; Qin et al. 2024), significantly enhancing recommendation performance and robustness by enabling the learning of higher-quality intent representations.

Figure 1: An example that illustrates the dynamic changes of a user's interaction sequence and intents.

While existing approaches achieve strong performance by modeling user intents, they often overlook the dynamic and evolving nature of these intents during interactions, resulting in suboptimal outcomes. Figure 1 illustrates the evolution of a user's interaction sequence, showing a shift in intent from playing to drinking, then to reading, followed by decorating, and finally returning to playing. When making recommendations based on the user sequence s_5 shown in the figure, it is crucial to account for these various intents and their transitions. Relying solely on s_5 as input may hinder the model's ability to fully capture the user's varying intents and the underlying transition patterns.

To effectively capture users' varying intents, we propose a novel framework named Intent Oriented Contrastive Learning for Sequential Recommendation (IOCLRec). We first dynamically segment user historical interaction sequence into several subsequences (e.g., s_i for $i \in [1, 5]$ as shown in

*Corresponding author.

Figure 1), each representing a coarse-grained intent at different interaction times. These coarse-grained intents serve as units for contrastive learning, and we design three contrastive learning modules accordingly. Specifically, in the fine-grained intent contrastive learning module, we bring closer the similar coarse-grained intent representations of different users, clustering all coarse-grained intents to obtain fine-grained intent representations. The single-intent and multi-intent contrastive learning modules employ three single-intent augmentation operators and two multi-intent augmentation operators, respectively, to enhance both dominant and diverse intents within the user behavior subsequences. Extensive experiments on four public datasets demonstrate that our approach effectively models user intent and improves recommendation performance.

2 Related Work

2.1 Sequential Recommendation

Sequential recommendation aims to model the dynamically changing interests of users based on their interaction sequences (Zheng et al. 2022; Du et al. 2023). Early works (He and McAuley 2016; Rendle, Freudenthaler, and Schmidt-Thieme 2010) primarily focused on capturing the transition patterns between items. With advancements in deep learning, various deep sequential recommendation models have emerged, including RNN-based (Hidasi et al. 2015) and CNN-based (Tang and Wang 2018) approaches. More recently, Transformer-based models have garnered significant attention. SASRec (Kang and McAuley 2018) utilized a unidirectional Transformer to model users' interaction sequences, while Bert4Rec (Sun et al. 2019) employed a bidirectional Transformer and a masked item prediction task to capture user behavior information from both directions, enhancing the performance over SASRec. LSAN (Li et al. 2021b) introduced a time-context-aware embedding, and (Fan et al. 2022) proposed a novel Wasserstein self-attention module to represent the relationship between items and their positions. Despite these advancements, these models remain challenged by the issue of data sparsity.

2.2 Contrastive SSL for Recommendation

Contrastive self-supervised learning trains an encoder by constructing different augmented views of an instance and maximizing the mutual information between them. This approach has been widely applied in recommendation systems to mitigate the issue of data sparsity. SGL (Wu et al. 2021) introduced node-level data augmentation operators on user-item graphs, generating multiple positive views of a node to improve recommendations. S3-Rec (Zhou et al. 2020) adopted a pre-training strategy to enhance data representation, using contrastive SSL during pre-training to maximize the mutual information between attributes and sequences. CL4SRec (Xie et al. 2022) created augmented views of sequences through three random data augmentation techniques: cropping, masking, and reordering. CoSeRec (Liu et al. 2021) improved upon CL4SRec by introducing two informative data operators that consider item correlations. TiCoSeRec (Dang et al. 2023) and UniRec (Liu, Wang, and Feng 2024) further incorporated temporal information to generate augmented views of sequences. DuoRec (Qiu et al. 2022) and REDA (Bian et al. 2022) took a different approach by generating augmented views at the model level rather than directly modifying the original interaction sequences. Unlike these methods, our approach further takes into account user intents when leveraging contrastive SSL.

2.3 User Intent for Recommendation

Recent studies have increasingly focused on improving recommendation performance by analyzing user intent (Li et al. 2019; Chen et al. 2020; Li et al. 2021a). DSSRec (Ma et al. 2020) introduced a seq2seq training strategy that leverages future interactions as supervision signals, incorporating intent variables to capture the mutual information between a user's historical and future interaction sequences. ICLRec (Chen et al. 2022) clustered user sequences to create intent prototypes, using random data augmentation to generate positive views. IOCRec (Li et al. 2023) applied global and local modules to model multiple intents within a user sequence, constructing positive views for each intent separately. ICSRec (Qin et al. 2024) extracted supervision signals from user interactions and applied contrastive learning to user subsequences based on these signals. Unlike these methods, our approach further accounts for the dynamic nature of user intents by designing three contrastive learning modules, each tailored to capture the user's intents at different stages of their interaction history.

3 Preliminary

3.1 Problem Definition

Let the sets of users and items be represented by $\mathcal{U}$ and $\mathcal{V}$, respectively. Each user $u \in \mathcal{U}$ is associated with an ordered sequence of items $S^u = \{v_1^u, v_2^u, \ldots, v_{|S^u|}^u\}$, where the length of the sequence is denoted by $|S^u|$, and $v_p^u \in \mathcal{V}$ for $1 \leq p \leq |S^u|$ indicates the item that user u interacted with at position p. The goal of sequential recommendation is to predict the next item in the sequence, $v_{|S^u|+1}^u$, which is formulated as follows:

$$\arg\max_{v_i \in \mathcal{V}} P(v_{|S^u|+1}^u = v_i \mid S^u). \tag{1}$$

3.2 Sequence Encoder

Given user sequence S^u, we utilize the Transformer architecture (Vaswani et al. 2017) as the sequence encoder to capture the dynamic evolution of user intents, formulated as:

$$\mathbf{h}^u = f_\theta(S^u), \tag{2}$$

where f_θ represents the sequence encoder, θ denotes the model parameters, and $\mathbf{h}^u$ is the sequence embedding of S^u. To optimize the encoder for next-item prediction, we employ the log-likelihood loss function, defined as follows:

$$\mathcal{L}_{\text{Rec}}(u,p) = -\log\sigma(\mathbf{h}_p^u \cdot \mathbf{e}_{v_{p+1}^u}) - \sum_{v_j \notin S^u} \log(1-\sigma(\mathbf{h}_p^u \cdot \mathbf{e}_{v_j})) \tag{3}$$

where $\mathcal{L}_{\text{Rec}}(u,p)$ denotes the loss score for the prediction at position p in sequence S^u, σ is the sigmoid function, $\mathbf{h}_p^u$

represents the predicted next item at position p (Kang and McAuley 2018), $\mathbf{e}_{v^u_{p+1}}$ and $\mathbf{e}_{v_j}$ denote the embeddings of item v^u_{p+1} and the sampled negative item v_j for S^u, respectively. The item embeddings are retrieved from the embedding table in f_θ, which is jointly optimized with the Transformer model.

4 Method

Figure 2 illustrates the overall framework of IOCLRec. IOCLRec first constructs a coarse-grained intent set for each user by dynamically segmenting their sequence. These subsequences are then fed into three contrastive learning modules to effectively learn user intent representations.

4.1 Item Correlation Modeling

Users tend to interact with similar items based on analogous intents. Therefore, before modeling user intent, it is essential to first model item correlation. Given the powerful language modeling capability of pre-trained language models, we first use the BERT model (Devlin et al. 2018) to capture the semantic information in the text descriptions of items. Specifically, we place a special symbol $[CLS]$ in front of the text description T_{v_i} of the given item v_i, which will serve as a special token, and its vector representation will contain the semantic information of the entire sentence. Then we feed the concatenated sequence into the BERT model:

$$t_i = \text{BERT}([[CLS], t_1, ..., t_{|T_{v_i}|}]), \tag{4}$$

where $t_i \in \mathbb{R}^d$ is the final hidden state of the $[CLS]$ symbol, which represents the semantic information of T_{v_i}, and d is the dimension size of BERT.

Given items v_i and v_j, their correlation score $\text{Cor}(v_i, v_j)$ combines three components: their collaborative filtering score C, their temporal interval T, and the cosine similarity R of their text representations t_i and t_j. The scores C, T, and R are all normalized to ensure consistency in the scoring mechanism. $\text{Cor}(v_i, v_j)$ is defined as:

$$\text{Cor}(v_i, v_j) = \phi(T, C, R), \tag{5}$$

$$\phi(x, y, z) = \frac{x + \Theta}{e^{(x+\Theta)/(\Gamma y + \Omega z)}}, \tag{6}$$

where Θ, Γ, and Ω are constants determined by the specific dataset and the types of variables involved. In this scoring mechanism, an increase in x or a decrease in y or z lowers $\phi(x, y, z)$. For an item v_i, if another item v_j has appeared in the same user sequence, v_i and v_j are considered neighbors. We refer to the top k neighbors of v_i as its k-neighbors, which are the items with the highest correlation scores to v_i.

4.2 Sequence Volatility Modeling

To measure the degree of variation in item correlation within a user sequence and apply it to subsequent sequence selection strategies, we introduce the concept of sequence volatility. For a given sequence $S^u = \{v^u_1, v^u_2, \ldots, v^u_{|S^u|}\}$, we define volatility of the sequence $V(S^u)$ as the standard deviation of the correlations between adjacent items:

$$V(S^u) = \sqrt{\frac{1}{|S^u| - 1} \sum_{i=1}^{|S^u|-1} \big(\text{Cor}(v^u_i, v^u_{i+1}) - \mu\big)^2}, \tag{7}$$

$$\mu = \frac{1}{|S^u| - 1} \sum_{i=1}^{|S^u|-1} \text{Cor}(v^u_i, v^u_{i+1}). \tag{8}$$

4.3 User Coarse-Grained Intent Set Construction

User sequences often reflect various distinct intents. To fully capture these diverse intents, we construct a coarse-grained intent set for each user, where each element in the set is a subsequence of the user's original sequence. Following (Qin et al. 2024), for a given sequence $S^u = \{v^u_1, v^u_2, \ldots, v^u_{|S^u|}\}$, the coarse-grained intent set is constructed as follows:

$$I(S^u) = \begin{cases} \{\{v^u_1, v^u_2\}, \{v^u_1, v^u_2, v^u_3\}, \\ \ldots, \{v^u_1, v^u_2, \ldots, v^u_{|S^u|}\}\} & |S^u| \le m \\ \\ I(s^u_m) \cup \{\{v^u_2, v^u_3, \ldots, v^u_{m+1}\}, \\ \ldots, \{v^u_{|S^u|-m+1}, \ldots, v^u_{|S^u|}\}\} & |S^u| > m, \end{cases} \tag{9}$$

where m denotes the maximum sequence length, s^u_m represents the subsequence of S^u consisting of the first m items.

4.4 Fine-Grained Intent Contrastive Learning

Coarse-Grained Intents Clustering. We encode all subsequences in the coarse-grained intent sets of all users via Eq. (2), denoted as $\cup_{u=1}^{|\mathcal{U}|} I(S^u)$. We then apply K-means clustering to these representations to obtain K intent centers, denoted as $\{\mathbf{c}_i\}_{i=1}^K$, which we refer to as fine-grained intent representations. By performing a query operation, we can identify the fine-grained intent representation corresponding to each coarse-grained intent.

Similar Intent Contrastive Learning. We design a strategy to identify comparable coarse-grained intents across different users, which helps the model better distinguish between various intents. For a subsequence s^u_i that ends with the item v^u_i, we sample from the coarse-grained intent sets of other users to create a set $Q(s^u_i)$. This set includes subsequences that end with the item v^u_i and where the preceding k_1 items are k_2-neighbors of the corresponding items in s^u_i.

If $|Q(s^u_i)| < r$ (where r is a parameter), we relax the conditions using the following strategies until $|Q(s^u_i)| \ge r$ or until all qualifying subsequences have been included:

- Gradually decrease the value of k_1 until $k_1 = 0$.
- Allow the last item of the subsequence to be a k_2-neighbor of v^u_i, while still ensuring the preceding k_1 items are k_2-neighbors of the corresponding items in s^u_i.

Next, we sample a subsequence s' from $Q(s^u_i)$ using a curriculum learning strategy. Specifically, for any subsequence $s_j \in Q(s^u_i)$, we assess its complexity c_j by evaluating both its volatility V and the number of changes D in the corresponding cluster centers over the last g epochs (where g is a parameter). Both V and D are normalized. The complexity c_j is calculated via Eq. (6):

$$c_j = \phi(0, V, D), \tag{10}$$

We simulate the human learning process, progressing from simple to complex. Each sequence s_j in $Q(s^u_i)$ is dy-

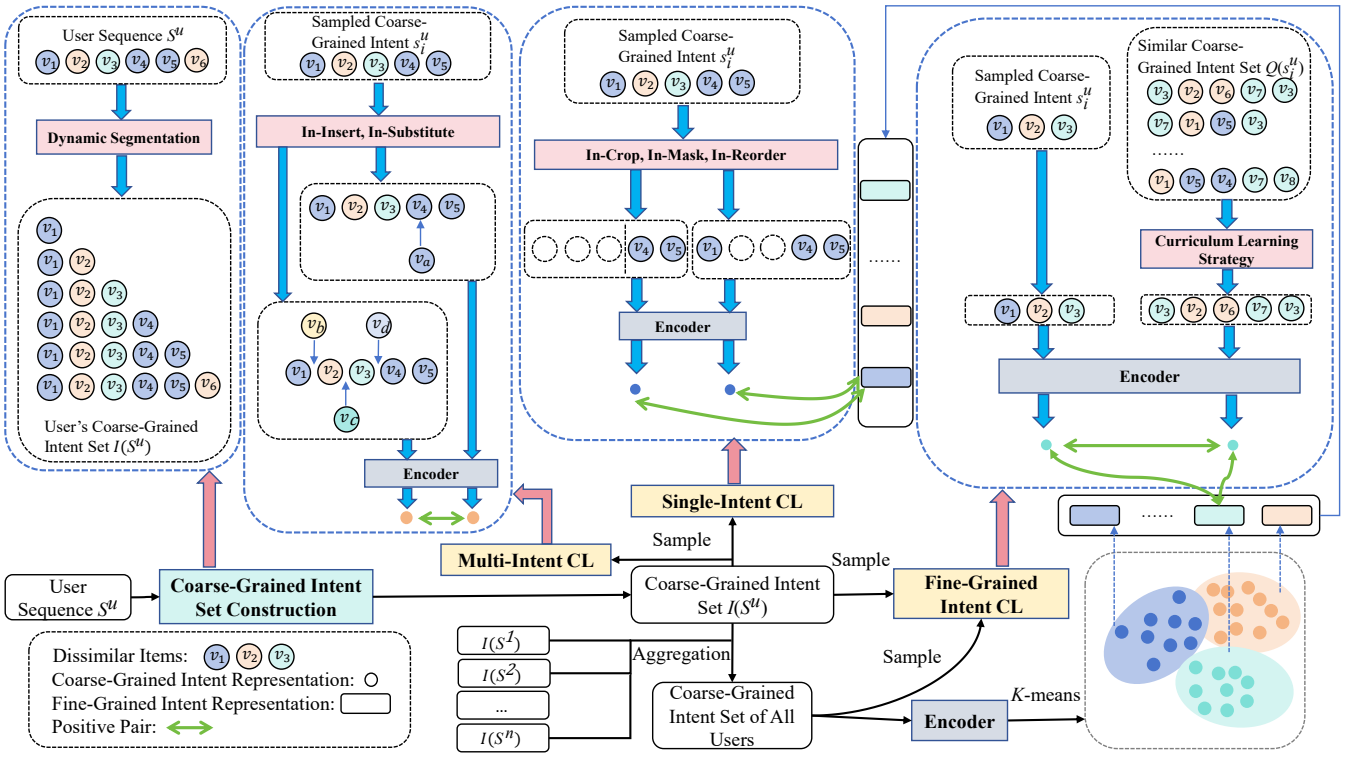


Figure 2: The model architecture of IOCLRec. The user sequence S^u is first segmented into multiple subsequences. These subsequences, representing the user's coarse-grained intents, serve as the units for three contrastive learning modules.

namically assigned a weight w_j as follows:

$$w_j = (1 - \frac{e}{e_{all}})\frac{c_{max} - c_j}{c_{max} - c_{min}} + \frac{e}{e_{all}}, \tag{11}$$

where e denotes the current epoch, e_{all} denotes the total number of epochs, c_{max} is the maximum complexity among all sequences in $Q(s_i^u)$, while c_{min} is the minimum.

After obtaining the sampled s', we use contrastive learning to reduce the distance between s_i^u and s' in the latent space. The contrastive loss is defined as follows:

$$\mathcal{L}_{\text{CoarseICL}} = \mathcal{L}_{\text{ICL}}(\mathbf{h}_1, \mathbf{h}_2), \tag{12}$$

where $\mathbf{h}_1$ and $\mathbf{h}_2$ are the encoded representations for s_i^u and s'respectively, and

$$\begin{aligned}\mathcal{L}_{\text{ICL}}(x_1, x_2) = &-\log\frac{\exp(\text{sim}(x_1, x_2))}{\sum_{x_i \notin \mathcal{F}} \exp(\text{sim}(x_1, x_i))} \\ &- \log\frac{\exp(\text{sim}(x_2, x_1))}{\sum_{x_i \notin \mathcal{F}} \exp(\text{sim}(x_2, x_i))},\end{aligned} \tag{13}$$

where $\text{sim}(\cdot)$ denotes the dot product, and $\mathcal{F}$ is constructed based on the False Negative Mitigation strategy (Liu et al. 2021; Qin et al. 2024). Specifically, $\mathcal{F}$ consists of the sequence representations in the mini-batch whose original sequences share the same ending item as the sequence represented by x_1 when x_2 is a sequence representation. Alternatively, if x_2 is a fine-grained intent representation, $\mathcal{F}$ includes the sequence representations in the mini-batch whose fine-grained intent representations match x_2.

Coarse-to-Fine Intent Contrastive Learning. Assuming $\mathbf{h}_1$ and $\mathbf{h}_2$ correspond to the fine-grained intent representations $\mathbf{c}_1$ and $\mathbf{c}_2$, respectively, we apply contrastive learning to reduce their distances in the latent space:

$$\mathcal{L}_{\text{FineICL}} = \mathcal{L}_{\text{ICL}}(\mathbf{h}_1, \mathbf{c}_1) + \mathcal{L}_{\text{ICL}}(\mathbf{h}_2, \mathbf{c}_2). \tag{14}$$

4.5 Single-Intent Contrastive Learning

The final item in each subsequence within a user's coarse-grained intent set serves as a crucial indicator of the user's primary intent during that interaction. Leveraging this signal, we propose three single-intent data operators for a given subsequence s_i^u.

In-Crop (IC). This operator selects a truncation position p from the sequence s_i^u, retains all items after v_p^u to form a new sequence, and ensures that this truncated sequence has a minimum length of l. To ensure that the intent represented by this new truncated sequence better reflects the intent associated with interacting with v_i^u, we first identify all subsequences that meet the length requirement. Then we calculate the average correlation score between each subsequence's items (excluding v_i^u itself) and v_i^u. The subsequence with the highest average correlation score is selected as the final result. This operator can be expressed as follows:

$$p = \arg\max_{p \in [1, i-l+1]} \frac{1}{i-p} \sum_{k=p}^{i-1} \text{Cor}(v_k^u, v_i^u). \tag{15}$$

In-Mask (IM). In this operator, $h_1 = \eta|s_i^u|$ items are selected from s_i^u and removed, where $\eta \in (0, 1)$ represents the mask proportion. To ensure the augmented sequence better reflects the intent associated with the user's interaction with v_i^u, we calculate the correlation scores of all remaining items in s_i^u with v_i^u via Eq. (5), generating a score sequence $O(s_i^u)$. We then sort $O(s_i^u)$ in ascending order (denoted as ascend$(O(s_i^u))$), and select the top h_1 indices as target positions for masking. This can be formulated as:

$$(p_1, p_2, \ldots, p_{h_1}) = \text{top-}h_1\text{-indices}(\text{ascend}(O(s_i^u))). \quad (16)$$

In-Reorder (IR). This operator first employs **IC** to extract the subsequence that most accurately reflects the user's current intent, then reorders the items in s_i^u based on their correlation with v_i^u. Specifically, **IC** is applied to s_i^u to obtain subsequence $s_{p:i}^u = \{v_p^u, v_{p+1}^u, \ldots, v_i^u\}$. Next, the correlation scores of all other items in $s_{p:i}^u$ with v_i^u are calculated via Eq. (5), resulting in a score sequence $O(s_{p:i}^u)$. This sequence is then sorted in ascending order (denoted as ascend$(O(s_{p:i}^u))$), and the indices of the sorted sequence are extracted (denoted as get-indices) to determine the sorted positions:

$$P_1 = \text{get-indices}(\text{ascend}(O(s_{p:i}^u))), \quad (17)$$

where P_1 is the set of sorted positions. Finally, the items in $s_{p:i}^u$ are reordered according to these sorted positions.

Contrastive Self-Supervision. Given a user sequence S^u, subsequences within $I(S^u)$ that have a length greater than $\epsilon|S^u|$ are selected as candidates for data augmentation, where ϵ controls the minimum length. Suppose s_i^u is a qualified subsequence, we randomly apply two single-intent data operators to s_i^u, yielding a pair of augmented sequences $(s_{i1}^{\tilde{u}}, s_{i2}^{\tilde{u}})$ and their corresponding representations $(\mathbf{h}_1^{\tilde{u}}, \mathbf{h}_2^{\tilde{u}})$. Assuming the fine-grain intent representation corresponding to subsequence s_i^u is $\mathbf{c}_i$, we use contrastive learning to minimize the distance between $(\mathbf{h}_1^{\tilde{u}}, \mathbf{h}_2^{\tilde{u}})$ and $\mathbf{c}_i$:

$$\mathcal{L}_{\text{SingleICL}} = \mathcal{L}_{\text{ICL}}(\mathbf{h}_1^{\tilde{u}}, \mathbf{c}_i) + \mathcal{L}_{\text{ICL}}(\mathbf{h}_2^{\tilde{u}}, \mathbf{c}_i). \quad (18)$$

4.6 Multi-Intent Contrastive Learning

We introduce two multi-intent data operators to enhance the transition and co-occurrence patterns of user intents for a given subsequence s_i^u.

In-Insert (II). This operator inserts the item most closely related to the items between which an intent transition occurs, thereby smoothing the user's intent transition. It selects $h_2 = \gamma|s_i^u|$ target positions and inserts a new item after the items at these positions, where $\gamma \in (0, 1)$ is the insertion rate. Specifically, for any position p_t ($t \in [1, i-1]$), we calculate the intent transition score via Eq. (6):

$$\text{Tran}(p_t) = \phi(\text{Sim}(p_t), \text{Rec}(p_t), \text{Intent}(p_t)), \quad (19)$$

where $\text{Sim}(p_t)$ denotes the correlation score between items v_t and v_{t+1}, and $\text{Rec}(p_t)$ represents the next-item prediction loss for subsequence s_t^u from the previous training epoch, calculated via Eq. (3). Both $\text{Sim}(p_t)$ and $\text{Rec}(p_t)$ are normalized. Intent$(\cdot)$ is a binary function, if s_t^u and s_{t+1}^u correspond to different fine-grained intents in the previous epoch, $\text{Intent}(p_t)$ is set to a constant c, otherwise it is set to 0.

Dataset	Beauty	Sports	Toys	ML-1M
#Users	22,363	35,598	19,412	6,040
#Items	12,101	18,357	11,924	3,416
#Interactions	198,502	296,337	167,597	999,611
#Average Length	8.9	8.3	8.6	165.4
Sparsity	99.92%	99.95%	99.93%	95.16%

Table 1: Statistics of the experimented datasets.

The intent transition score is calculated for each position, forming $O(s_i^u) = \{\text{Tran}(p_1), \text{Tran}(p_2), \ldots, \text{Tran}(p_{i-1})\}$. These scores are then sorted in descending order (denoted as descend$(O(s_i^u))$), and the top h_2 indices are selected:

$$P_2 = \text{top-}h_2\text{-indices}(\text{descend}(O(s_i^u))), \quad (20)$$

where P_2 denotes the set of target positions. For each $p_t \in P_2$, we identify the item within the common neighbor set of v_t and v_{t+1} that has the highest total correlation score with both v_t and v_{t+1}. We insert this item between v_t and v_{t+1}.

In-Substitute (IS). This operator also calculates the intent transition score for each position in the sequence. However, unlike **II**, this operator selects the $h_3 = \delta|s_i^u|$ positions with the lowest intent transition scores for replacement, where $\delta \in (0, 1)$ is the substitute rate. The item selection strategy for this operator is the same as that of **II**.

Contrastive Self-Supervision. Given a user sequence S^u, subsequences within $I(S^u)$ that have a length greater than $\omega|S^u|$ are selected as candidates for data augmentation, where ω controls the minimum length. Suppose s_i^u is a qualified subsequence, we apply the above two multi-intent data operators to s_i^u to obtain a pair of augmented sequences $(s_{i1}^{\tilde{u}}, s_{i2}^{\tilde{u}})$ and their representations $(\mathbf{h}_1^{\tilde{u}}, \mathbf{h}_2^{\tilde{u}})$. These augmented sequences are considered a positive pair, and the contrastive loss is computed as follows:

$$\mathcal{L}_{\text{MultiICL}} = \mathcal{L}_{\text{ICL}}(\mathbf{h}_1^{\tilde{u}}, \mathbf{h}_2^{\tilde{u}}). \quad (21)$$

4.7 Multi-Task Training

We use a multi-task learning paradigm to jointly optimize the main sequential prediction task and other auxiliary learning objectives, which can be formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{Rec}} + \lambda\mathcal{L}_{\text{CoarseICL}} + \alpha(\mathcal{L}_{\text{FineICL}} + \mathcal{L}_{\text{SingleICL}}) + \beta\mathcal{L}_{\text{MultiICL}}, \quad (22)$$

where λ, α and β are hyper-parameters that need to be tuned.

5 Experiments

5.1 Experimental Setting

Datasets. We conduct experiments on four public datasets. **Sports**, **Beauty** and **Toys** are three subcategories of Amazon review data introduced in (McAuley et al. 2015). MovieLens-1M (Harper and Konstan 2015) is a dataset containing users' behavior logs on movies, denoted as **ML-1M**.

Following (Chen et al. 2022; Xie et al. 2022), we only retain the '5-core' datasets, where each user and item has at least 5 interactions. The statistics of all processed datasets are shown in Table 1.

Dataset	Metric	BPR	Caser	SASRec	BERT4Rec	CoSeRec	DuoRec	ICLRec	IOCRec	ICSRec	Ours	Improve
Sports	HR@10	0.0205	0.0260	0.0330	0.0331	0.0433	0.0478	0.0433	0.0452	0.0558	**0.0628**	12.54%
	HR@20	0.0324	0.0402	0.0498	0.0546	0.0679	0.0711	0.0642	0.0679	0.0791	**0.0890**	12.52%
	NDCG@10	0.0098	0.0135	0.0172	0.0176	0.0244	0.0235	0.0232	0.0214	0.0329	**0.0363**	10.33%
	NDCG@20	0.0136	0.0179	0.0217	0.0231	0.0295	0.0298	0.0286	0.0281	0.0389	**0.0429**	10.28%
Beauty	HR@10	0.0358	0.0341	0.0624	0.0587	0.0721	0.0821	0.0741	0.0769	0.0944	**0.1092**	15.68%
	HR@20	0.0593	0.0643	0.0903	0.0966	0.1033	0.1222	0.1077	0.1145	0.1295	**0.1504**	16.14%
	NDCG@10	0.0187	0.0225	0.0339	0.0304	0.0389	0.0421	0.0397	0.0391	0.0572	**0.0657**	14.86%
	NDCG@20	0.0232	0.0298	0.0385	0.0395	0.0472	0.0527	0.0488	0.0488	0.0660	**0.0761**	15.30%
Toys	HR@10	0.0185	0.0292	0.0668	0.0517	0.0742	0.0927	0.0828	0.0802	0.1041	**0.1222**	17.39%
	HR@20	0.0316	0.0436	0.0946	0.0756	0.1029	0.1286	0.1149	0.1129	0.1365	**0.1620**	18.68%
	NDCG@10	0.0101	0.0136	0.0341	0.0295	0.0426	0.0481	0.0476	0.0382	0.0649	**0.0747**	15.10%
	NDCG@20	0.0137	0.0207	0.0435	0.0353	0.0494	0.0578	0.0561	0.0465	0.0731	**0.0848**	16.01%
Ml-1M	HR@10	0.0409	0.1389	0.1776	0.2167	0.1802	0.2946	0.2168	0.2604	0.3231	**0.3457**	6.99%
	HR@20	0.0733	0.2176	0.2663	0.3224	0.2766	0.3894	0.3237	0.3719	0.4345	**0.4541**	4.51%
	NDCG@10	0.0206	0.0649	0.0910	0.1008	0.0891	0.1683	0.1119	0.1422	0.1937	**0.2048**	5.73%
	NDCG@20	0.0279	0.0848	0.1092	0.1317	0.1183	0.1946	0.1386	0.1691	0.2211	**0.2321**	4.98%

Table 2: Performance comparison across different methods. For each row, the best score is highlighted in bold, and the second-best score is underlined. The final column shows the relative improvement over the top baseline result.

Evaluation Metrics. We follow (Wang et al. 2019) to rank the whole item set without negative sampling. The evaluation metrics include Hit Ratio@N (HR@N), and Normalized Discounted Cumulative Gain@N (NDCG@N). We report HR and NDCG with N $\in \{10, 20\}$.

Baseline Models. We pick four groups of baseline models:

- **Non-sequential model**: BPR (Rendle et al. 2012).
- **General sequential models**: Caser (Tang and Wang 2018) and SASRec (Kang and McAuley 2018).
- **Sequential models with SSL**: BERT4Rec (Sun et al. 2019), CoSeRec (Liu et al. 2021) and DuoRec (Qiu et al. 2022).
- **Intent-guided sequential models**: ICLRec (Chen et al. 2022), IOCRec (Li et al. 2023) and ICSRec (Qin et al. 2024).

Implementation Details. All baseline models are implemented based on public resources or codes provided by the respective authors. Our method is implemented in PyTorch. We set the number of self-attention blocks and attention heads to 2, the embedding dimension to 64. The batch size is set to 256. We use the Adam optimizer (Kingma and Ba 2014) with a learning rate of 0.001. Each data operator's sampling ratio (i.e., η, γ, and δ) is varied within the range [0.1, 0.9] (stepping by 0.1). Parameters ϵ and ω range from 0.5 to 1 (stepping by 0.1). Parameters λ, α, β and the dropout rate are set within the range $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. The number of clusters is chosen from $\{64, 128, 256, 512, 1024\}$. The values for k_1, k_2, g, l, r and m are set to 2, 5, 10, 3, 20 and 50, respectively. All experiments are conducted on a single Tesla V100 GPU.

5.2 Performance Comparison

We evaluate our method against all baseline models across various datasets, with the results presented in Table 2. Our analysis leads to the following conclusions:

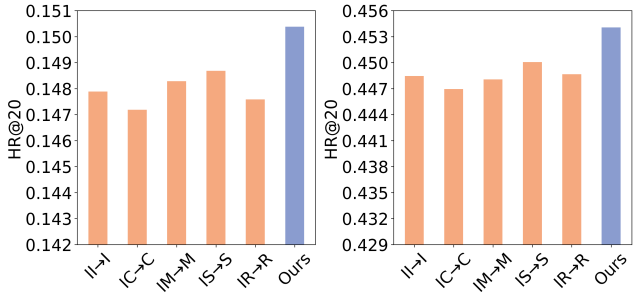
- The BPR model underperforms compared to general sequential models. SASRec, utilizing the attention mechanism, outperforms Caser. BERT4Rec, trained with a masked item prediction task, surpasses SASRec in most cases. CoSeRec and DuoRec further improve learned sequence representations through contrastive learning, yielding better results than SASRec.
- ICLRec and IOCRec use contrastive SSL to capture user intent, outperforming most baselines. ICSRec introduces an intent supervision signal, enabling even better learning of user intent and outperforming all other baselines.
- By constructing intent-level contrastive learning tasks, our method consistently outperforms all baseline models across different datasets on all metrics. Specifically, it achieves a performance improvement of 10.28-18.68% on the three sparse datasets and 4.98-6.99% on the dense dataset ML-1M, measured in terms of HR and NDCG. The relatively smaller improvement on ML-1M may be attributed to the generally longer user sequence lengths in this dataset compared to the others, which leads to more diverse user intentions and increases the difficulty for the model to effectively learn distinct user intents.

5.3 Ablation Study

IOCLRec with Other Variants. To evaluate the contribution of each component in IOCLRec, we conduct ablation experiments. Table 3 summarizes the NDCG@10 performance of IOCLRec and its variants across four datasets. In this table, (A) represents the full IOCLRec model, while (B) through (E) represent variants where $\mathcal{L}_{\text{CoarseICL}}$, $\mathcal{L}_{\text{FineICL}}$, $\mathcal{L}_{\text{SingleICL}}$, and $\mathcal{L}_{\text{MultiICL}}$ in Eq. (22) are set to 0, respectively, with all other components remaining unchanged. The results show that IOCLRec outperforms its variants on all datasets,

Model	Dataset			
	Sports	Beauty	Toys	ML-1M
(A)IOCLRec	**0.0363**	**0.0657**	**0.0747**	**0.2048**
(B)w/o CoarseICL	0.0325	0.0591	0.0682	0.1975
(C)w/o FineICL	0.0333	0.0629	0.0723	0.2006
(D)w/o SingleICL	0.0342	0.0611	0.0696	0.1993
(E)w/o MultiICL	0.0347	0.0628	0.0704	0.2012

Table 3: The NCDG@10 performance achieved by IOCLRec variants on four datasets.



(a) Beauty (b) ML-1M

Figure 3: Ablation study of data operators on two datasets.

indicating that all components are effective.

Effect of Data Operators. Figure 3 illustrates how each data operator affects overall HR @20 performance. We replace each data operator with the original operators introduced in CoSeRec (denoted by, e.g., IC→C) while keeping everything else the same. When our operators are swapped for traditional ones, recommendation performance drops. Due to similar trends and space limitations, we have not included results for the Toys and Sports datasets.

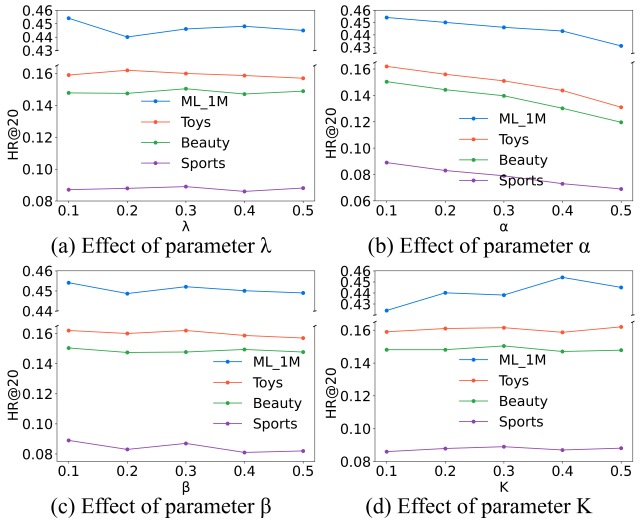
5.4 Hyper-parameter Sensitivity

Impact of λ, α, and β. Figures 4(a), 4(b), and 4(c) show how recommendation performance varies with different values of λ, α, and β. The best performance for the Beauty and Sports datasets is with $\lambda = 0.3$, $\alpha = 0.1$, and $\beta = 0.1$. For the Toys dataset, it's $\lambda = 0.2$, $\alpha = 0.1$, and $\beta = 0.1$. For ML-1M, the best values are $\lambda = 0.1$, $\alpha = 0.1$, and $\beta = 0.1$.

Impact of K. Figure 4(d) shows the performance for different numbers of clusters K. The optimal K values are 256 for Beauty and Sports, 1024 for Toys, and 512 for ML-1M.

5.5 User Representation Case Study

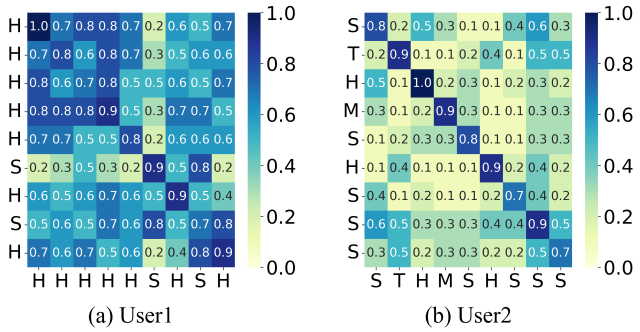
We conduct a case study to evaluate the effectiveness of our method in modeling user intents. Specifically, we randomly select two user sequences from the Beauty dataset and compute the product of their representations with their transposes, producing two matrices. These matrices illustrate the similarities between item representations within each user sequence, as depicted in Figure 5. In this figure, each item is labeled with its corresponding category: H for Hair Care, S for Skin Care, T for Tools & Accessories, and M for



(a) Effect of parameter λ (b) Effect of parameter α

(c) Effect of parameter β (d) Effect of parameter K

Figure 4: Performance of IOCLRec on HR@20 with varying hyperparameters (λ, α, β, and K).



(a) User1 (b) User2

Figure 5: Item representation similarity heatmap.

Makeup. User1's sequence primarily consists of items from a single category, and our method successfully captures the user's stable intents, resulting in high similarity among item representations within the sequence. In contrast, User2's sequence contains items spanning multiple categories, which our method effectively identifies, leading to lower similarity among item representations within the sequence.

6 Conclusion

In this paper, we introduce IOCLRec, a novel sequence recommendation model designed to effectively capture evolving user intents. IOCLRec utilizes subsequences of user interactions as units for three distinct contrastive learning modules. The fine-grained intent contrastive learning module enhances intent representations by identifying and clustering similar subsequences. Meanwhile, the single-intent and multi-intent contrastive learning modules use three single-intent augmentation operators and two multi-intent augmentation operators, respectively, to enhance both dominant and diverse intents within user behavior subsequences. Extensive experiments on four public datasets validate the effectiveness of our proposed method.

References

Bian, S.; Zhao, W. X.; Wang, J.; and Wen, J.-R. 2022. A relevant and diverse retrieval-enhanced data augmentation framework for sequential recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2923–2932.

Chen, W.; Ren, P.; Cai, F.; Sun, F.; and de Rijke, M. 2020. Improving end-to-end sequential recommendations with intent-aware diversification. In *Proceedings of the 29th ACM international conference on information & knowledge management*, 175–184.

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.

Dang, Y.; Yang, E.; Guo, G.; Jiang, L.; Wang, X.; Xu, X.; Sun, Q.; and Liu, H. 2023. TiCoSeRec: Augmenting data to uniform sequences by time intervals for effective recommendation. *IEEE Transactions on Knowledge and Data Engineering*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Du, J.; Ye, Z.; Guo, B.; Yu, Z.; and Yao, L. 2023. Idnp: Interest dynamics modeling using generative neural processes for sequential recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, 481–489.

Fan, Z.; Liu, Z.; Wang, Y.; Wang, A.; Nazari, Z.; Zheng, L.; Peng, H.; and Yu, P. S. 2022. Sequential recommendation via stochastic self-attention. In *Proceedings of the ACM web conference 2022*, 2036–2047.

Harper, F. M.; and Konstan, J. A. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4): 1–19.

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.

Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.

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

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, C.; Liu, Z.; Wu, M.; Xu, Y.; Zhao, H.; Huang, P.; Kang, G.; Chen, Q.; Li, W.; and Lee, D. L. 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.

Li, H.; Wang, X.; Zhang, Z.; Ma, J.; Cui, P.; and Zhu, W. 2021a. Intention-aware sequential recommendation with structured intent transition. *IEEE Transactions on Knowledge and Data Engineering*, 34(11): 5403–5414.

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, Y.; Chen, T.; Zhang, P.-F.; and Yin, H. 2021b. Lightweight self-attentive sequential recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 967–977.

Liu, Y.; Wang, Y.; and Feng, C. 2024. UniRec: A Dual Enhancement of Uniformity and Frequency in Sequential Recommendations. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 1483–1492.

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

Ma, J.; Zhou, C.; Yang, H.; Cui, P.; Wang, X.; and Zhu, W. 2020. Disentangled self-supervision in sequential recommenders. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 483–491.

McAuley, J.; Targett, C.; Shi, Q.; and Van Den Hengel, A. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, 43–52.

Qin, X.; Yuan, H.; Zhao, P.; Liu, G.; Zhuang, F.; and Sheng, V. S. 2024. Intent Contrastive Learning with Cross Subsequences for Sequential Recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, 548–556.

Qiu, R.; Huang, Z.; and Yin, H. 2021. Memory augmented multi-instance contrastive predictive coding for sequential recommendation. In *2021 IEEE International Conference on Data Mining (ICDM)*, 519–528. IEEE.

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.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*.

Rendle, S.; Freudenthaler, C.; and Schmidt-Thieme, L. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, 811–820.

Roy, D.; and Dutta, M. 2022. A systematic review and research perspective on recommender systems. *Journal of Big Data*, 9(1): 59.

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.

Tang, J.; and Wang, K. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*, 565–573.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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, Y.; Wang, X.; Huang, X.; Yu, Y.; Li, H.; Zhang, M.; Guo, Z.; and Wu, W. 2024. Intent-aware recommendation via disentangled graph contrastive learning. *arXiv preprint arXiv:2403.03714*.

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, X.; Xiong, Y.; Zhang, Y.; Jiao, Y.; and Zhang, J. 2023. Dual Intents Graph Modeling for User-centric Group Discovery. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2716–2725.

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.

Zhang, K.; Qian, H.; Cui, Q.; Liu, Q.; Li, L.; Zhou, J.; Ma, J.; and Chen, E. 2021a. Multi-interactive attention network for fine-grained feature learning in ctr prediction. In *Proceedings of the 14th ACM international conference on web search and data mining*, 984–992.

Zhang, K.; Qian, H.; Liu, Q.; Zhang, Z.; Zhou, J.; Ma, J.; and Chen, E. 2021b. Sifn: A sentiment-aware interactive fusion network for review-based item recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 3627–3631.

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, 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.