Multi-Intention Oriented Contrastive Learning for Sequential Recommendation

Xuewei Li
College of Intelligence and Computing, Tianjin University
Tianjin, China
lixuewei@tju.edu.cn

Aitong Sun
College of Intelligence and Computing, Tianjin University
Tianjin, China
aitongsun@tju.edu.cn

Mankun Zhao
College of Intelligence and Computing, Tianjin University
Tianjin, China
zmk@tju.edu.cn

Jian Yu
College of Intelligence and Computing, Tianjin University
Tianjin, China
yujian@tju.edu.cn

Kun Zhu
Tianjin International Engineering Institute, Tianjin University
Tianjin, China
2019229048@tju.edu.cn

Di Jin
College of Intelligence and Computing, Tianjin University
Tianjin, China
jindi@tju.edu.cn

Mei Yu*
College of Intelligence and Computing, Tianjin University
Tianjin, China
yumei@tju.edu.cn

Ruiguo Yu
College of Intelligence and Computing, Tianjin University
Tianjin, China
rgyu@tju.edu.cn

ABSTRACT

Sequential recommendation aims to capture users' dynamic preferences, in which data sparsity is a key problem. Most contrastive learning models leverage data augmentation to address this problem, but they amplify noises in original sequences. Contrastive learning has the assumption that two views (positive pairs) obtained from the same user behavior sequence must be similar. However, noises typically disturb the user's main intention, which results in the dissimilarity of two views.

To address this problem, in this work, we formalize the denoising problem by selecting the user's main intention, and apply contrastive learning for the first time under this topic, i.e., we propose a novel framework, namely Multi-**I**ntention **O**riented **C**ontrastive Learning **Rec**ommender (IOCRec). In order to create high-quality views with intent-level, we fuse local and global intentions to unify sequential patterns and intent-level self-supervision signals. Specifically, we design the sequence encoder in IOCRec which includes three modules: local module, global module and disentangled module. The global module can capture users' global preferences, which is independent of the local module. The disentangled module can obtain multi-intention behind global and local representations. From a fine-grained perspective, IOCRec separates different intentions to guide the denoising process. Extensive experiments on four widely-used real datasets demonstrate the effectiveness of our new method for sequential recommendation.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Sequential Recommendation, Contrastive Learning, Multi-Intention Modeling

ACM Reference Format:
Xuewei Li, Aitong Sun, Mankun Zhao, Jian Yu, Kun Zhu, Di Jin, Mei Yu, and Ruiguo Yu. 2023. Multi-Intention Oriented Contrastive Learning for Sequential Recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM '23), February 27–March 3, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3539597.3570411

1 INTRODUCTION

The goal of sequential recommendation (SR) is to capture users' dynamic preferences from their history behaviors, which is able to accurately make a next-item recommendation [4, 6, 9, 19, 27, 44, 47]. Pioneering work [30] adopts Markov chains to learn sequence relationships. With the prosperity of deep neural network, CNN-based [33] and RNN-based models [13, 39] have become two mainstreams to distill preference information for SR. In addition, attention-based models can effectively learn users' preferences by estimating an importance weight for each item [18, 29]. However, when the training data is limited, these methods may fail to infer appropriate user representations. Recently, inspired by Self-Supervised Learning (SSL), Contrastive Learning (CL) is gradually applied to SR. The CL mainly contains two key components: data augmentation for

*Corresponding author.

WSDM '23, February 27–March 3, 2023, Singapore, Singapore.

ACM ISBN 978-1-4503-9407-9/23/02…$15.00
https://doi.org/10.1145/3539597.3570411

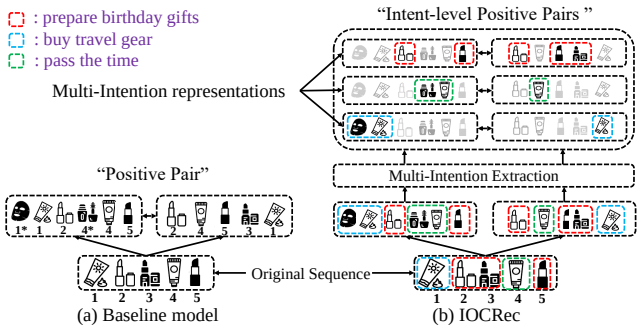


Figure 1: A case of contrastive learning strategies comparing baseline model and our new method IOCRec.

enhancing training data and contrastive losses for enhancing self-supervision signals. Therefore, CL-based models can alleviate the aforementioned issues in SR.

Though CL-based approaches mitigate data sparsity problem, they may amplify noises in the original sequence. In practice, the user selects each item based on an underlying intention, which can be understood as a subjective reason for the interaction (e.g. preparing birthday gifts, buying travel gear, passing the time, etc.). If the current intention of a user is to prepare birthday gifts, then items related to other historical intentions are regarded as noises. In the CL, each original sequence is transformed into two views (i.e., positive pairs), which are required to be similar. However, actual user intentions in the two views may not be similar since they are obscure and highly entangled. Taking Figure 1(a) as an example, in the original sequence (i.e., 1, 2, 3, 4, 5), the current intention of a user is to select a suitable *lipstick* (i.e., 2, 3, 5) to prepare birthday gifts. Meanwhile, noisy behaviors (i.e., 1, 4) commonly exist in a user's interaction history, worsening the extraction of user's current main intention. After data augmentation, in one view, noisy behaviors may introduce items that the user does not care about at the current moment, such as *facial mask* (i.e., 1*) and *mascara* (i.e., 4*). Moreover, the other view also modifies sequential patterns (i.e., 2, 4, 5, 3, 1), which changes the user's current intention to prepare birthday gifts. Therefore, two views without considering multi-intention conditions differ significantly, which limits the performance of CL in SR.

In this paper, we formalize the denoising problem by selecting the user's current main intention, our core idea is shown in Figure 1(b). To improve the incorporation of CL into the SR task, we propose a framework called Multi-**I**ntention **O**riented **C**ontrastive Learning **Rec**ommender (IOCRec), which mainly includes three key components: **(1) Sequence encoder integrating global and local intention representations.** To be specific, we disentangle multi-intention behind the global and local representations in the latent space. From a global perspective, especially in augmented sequences, it is difficult to distinguish noisy information in one sequence. So we capture more semantic information from all sequences to better distinguish current noisy items. From a local perspective, the local intention representations are inferred solely based on individual sequence representation, and the local representation tends to focus on more recent activities in one sequence. **(2) Multi-intention oriented CL module.** We create high-quality views by extracting multi-intention representations from augmented sequences. A novel design of the sequence encoder can better create intent-level positive pairs. In this way, IOCRec granularly separates different intentions of the same user, so as to select the user's current main intention in the denoising process. **(3) A multi-task training strategy.** IOCRec can jointly optimize the CL objective and SR objective effectively. Furthermore, in the SR task, we select the current main intention to predict the next item. In the CL task, we maximize the consistency of shared intentions and the diversity of different intentions to jointly optimize the parameters of the sequence encoder.

Our main contributions are summarized as follows: (1) To the best of our knowledge, this is the first work to apply intent-level contrastive learning for denoising problem of the SR task. (2) We propose a sequence encoder integrating global and local intention representations to select the current main intention. (3) Empirical results demonstrate that the process of sequence denoising is feasible in CL and our framework achieves state-of-the-art performance on four real-world challenging datasets.

2 PRELIMINARIES

2.1 Problem Definition

We denote user and item sets as $\mathcal{U}$ and $\mathcal{V}$ respectively. Given a user $u \in \mathcal{U}$, the user behavior s_u is associated with a sequence of items $s_u = [v_1, v_2, \cdots, v_L]$, where $v \in \mathcal{V}$, L denotes the total number of items in a sequence. The purpose of sequence recommendation is generally assessed as to predict next-item $v_{|s_u|+1}$, which is formulated as follows:

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

which is expressed as computing the probability of all candidate items and selecting the highest score to recommend.

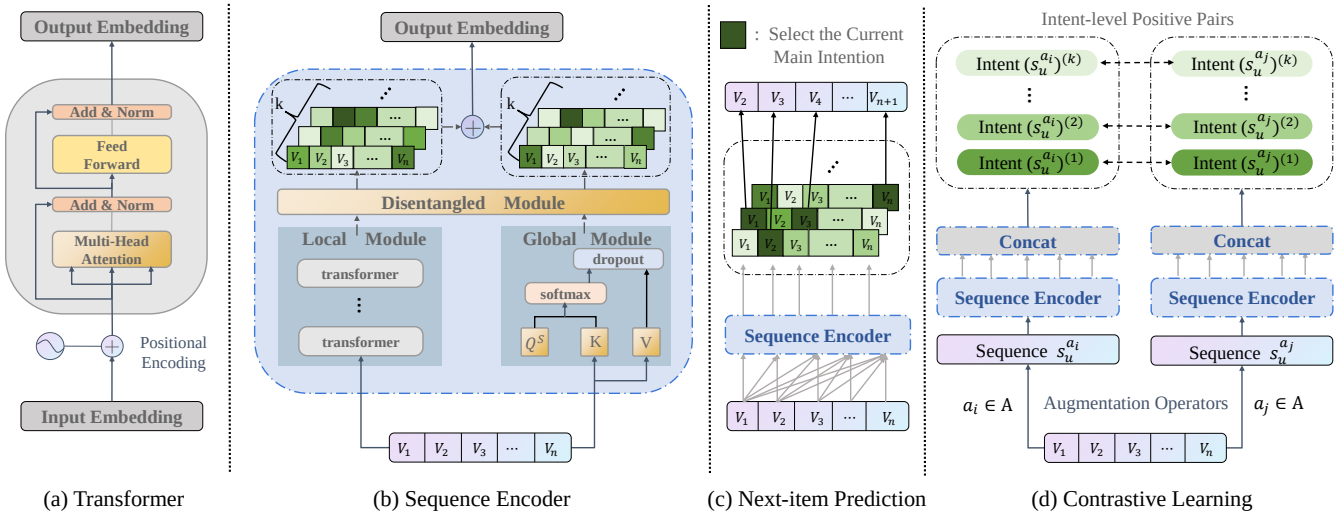
2.2 Multi-Intention Definition

Users may have diverse intentions, i.e. the subjective reasons why users interact with items (e.g., purchasing travel gear, preparing for birthday gifts, passing the time, etc.). We aim to preserve the user's intentions under k latent categories, namely $\mathbf{c}_u = [\mathbf{c}_u^{(1)}; \mathbf{c}_u^{(2)}; \ldots; \mathbf{c}_u^{(k)}] \in \mathbb{R}^d$. The sequence encoder ϕ projects each historical item of s_u into k latent spaces with a certain probability expressed as $\phi_{\boldsymbol{\theta}}(s_u) = P_{\boldsymbol{\theta}}(s_u, \mathbf{c}_u)$, where $\boldsymbol{\theta}$ is the set that contains all the trainable parameters, and then estimates the probability that the user u will click the i^{th} item by measuring the similarity between the user's intention representation $\phi_{\boldsymbol{\theta}}(s_u)$ and the i^{th} item's representation in the vector space.

2.3 Data Augmentation Operators

Five data-level augmentation operators introduced by [21, 43] are included in this paper:

- **Crop (C):** Randomly select a continuous sub-sequence $L_C = \lceil \eta * |s_u| \rceil$, where the L_C is controlled by a hyperparameter η. The randomly cropped sub-sequence starting from position



(a) Transformer (b) Sequence Encoder (c) Next-item Prediction (d) Contrastive Learning

Figure 2: Overall framework. (a) illustrates the structure of Transformer. (b) presents the structure of Sequence Encoder. (c) predicts the next item based on selecting the current main intention. (d) demonstrates CL of a sequence. The augmentation operators set A = {C, M, R, S, I}, the details see Section 2.3, it first augments a sequence as positive pair with random select two augmentation operators a_i,a_j from set A. Then, it encodes the sequence by concatenating embedding outputs from Sequence Encoder. Finally, it maximizes the agreement between intent-level positive pairs.

c can be formulated as:

$$s_u^{\text{Crop}} = \left[v_c, v_{c+1}, \ldots, v_{c+L_c-1}\right], \tag{2}$$

- **Mask (M):** Randomly mask $l = \lceil \gamma * |s_u| \rceil$ items in an original sequence, where l is the total number of items selected to mask. This mask method can be formulated as:

$$s_u^{\text{Mask}} = \left[\widehat{v}_1, \widehat{v}_2, \ldots, \widehat{v}_{|s_u|}\right], \tag{3}$$

where $\widehat{v}_i$ is the 'mask' if v_i is a selected item, otherwise $\widehat{v}_i = v_i$.

- **Reorder (R):** Randomly shuffle a continuous sub-sequence $[v_r, v_{r+1}, \ldots, v_{r+L_R-1}]$, where $L_R = \lceil \mu * |s_u| \rceil$ is the length of sub-sequence, the reordered sequence can be formulated as:

$$s_u^{\text{Reorder}} = \left[v_1, \ldots, \widehat{v}_i, \ldots, \widehat{v}_{i+L_R-1}, \ldots, v_{|s_u|}\right], \tag{4}$$

- **Substitute (S):** Randomly select m different indices $\{idx_1, ..., idx_m\}$, where $idx_i \in |s_u|$, and replace each with the most similar item depending on the selected indices, where $m = \lceil \alpha * |s_u| \rceil$, the substituted sequence can be formulated as:

$$s_u^{\text{Substitute}} = \left[v_1, v_2, \ldots, \widehat{v}_{idx_i}, \ldots, v_{|s_u|}\right], \tag{5}$$

where $\widehat{v}_{\text{idx}_i}$ replaces v_{idx_i}, and we employ ItemCF-IUF[1] to measure item correlations.

- **Insert (I):** Randomly select m different indices, and insert the most similar item at those indices, where $m = \lceil \beta * |s_u| \rceil$, the inserted sequence can be formulated as:

$$s_u^{\text{Insert}} = \left[v_1, v_2, \ldots, \widehat{v}_{\text{idx}_i}, v_{\text{idx}_i}, \ldots, v_n\right]. \tag{6}$$

where $\widehat{v}_{\text{idx}_i}$ is the most similarity to v_{idx_i} by ItemCF-IUF[1].

3 METHODOLOGY

In this section, we will introduce the details of our proposed IOCRec. The overall framework is shown in Figure 2.

3.1 Local Module

The Transformer [36] facilitates the progress of SR since it is a strong encoder structure. To capture the influence of the position, we add a learnable position embedding matrix $\mathbf{P} = [p_1; p_2; \ldots; p_L] \in \mathbb{R}^{L\times d}$ to the embedding matrix $\mathbf{E} \in \mathbb{R}^{L\times d}$ of the original sequence s_u, so the input embedding is represented as $\mathbf{E}_P^{(0)}$. Then we feed the input embedding into a series of stacked Transformer structures as our local module. The output of the l-th Transformer structure can then be viewed as follows:

$$\mathbf{E}_P^{(l)} = Local^{(l)}\left(\mathbf{E}_P^{(l-1)}\right), l \in \{1, 2, \ldots\}, \tag{7}$$

We illustrate the structure of a Transformer in Figure 2(a), which consists of two components: *Multi-Head Attention* (MHA) and position-wise *Feed-Forward Network* (FFN). The key parts are briefly summarized as follows:

$$\begin{aligned} \text{MHA}\left(\mathbf{E}_P^{(l)}\right) &= \text{concat}\left(\text{head}_1; \cdots; \text{head}_h\right)\mathbf{W}^O, \\ \text{head}_i\left(\mathbf{E}_P^{(l)}\right) &= \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d/h}}\right)\mathbf{V}, \\ Local\left(\mathbf{E}_P^{(l)}\right) &= \left[\text{FFN}\left(\mathbf{E}_{P1}^{(l)}\right)^\top; \cdots; \text{FFN}\left(\mathbf{E}_{Pn}^{(l)}\right)^\top\right]. \end{aligned} \tag{8}$$

where $\mathbf{Q} = \mathbf{E}_P\mathbf{W}_Q, \mathbf{K} = \mathbf{E}_P\mathbf{W}_K$, and $\mathbf{V} = \mathbf{E}_P\mathbf{W}_V$ with $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d\times d/h}$ are the projected query, key and value matrices respectively to improve the flexibility. $\mathbf{W}^O \in \mathbb{R}^{d\times d}$ and the h is the number of $\mathbf{E}_P^{(0)}$ into the subspace using different linear projections. The factor

$\sqrt{d/h}$ in this attention module is the scale factor to avoid large values of the inner product. Position-wise FFN applies at each position of the above sub-layer's output with shared learnable parameters. Finally, we take the output matrix $\mathbf{E}_P^{(l)}$ from the top Transformer structure as the local representation.

3.2 Global Module

With the stacking of the transformer structures, the local representation may pay attention to the user's more recent interactions. However, in augmented sequences, a large amount of noise information will be introduced, and it is difficult to capture the main intention from the current sequence. So we introduce a learnable query matrix $\mathbf{Q}^S \in \mathbb{R}^{L\times d}$ shared by all sequences, so that more available information can be utilized for the prediction at each step in the sequence during training. As seen in Figure 2(b), the global module computation is defined as:

$$Global(\mathbf{E}) = \text{Dropout}(\text{softmax}\left(\mathbf{Q}^S\left(\mathbf{E}\mathbf{W}'_K\right)^T\right)\mathbf{E}\mathbf{W}'_V). \tag{9}$$

where $\mathbf{E} \in \mathbb{R}^{L\times d}$ is the input matrix of the original sequence s_u, $\mathbf{W}'_K, \mathbf{W}'_V \in \mathbb{R}^{d\times d}$, which are projection matrices to be learned, similar to $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$. Note that besides the query condition, the embedding layer in Eq.(9) also ignores that the position information $\mathbf{P}$. It is worth mentioning that in our case, the global representations of the sequence is adopted the same to all steps, which means that $\mathbf{Q}^S$ is updated after learning semantic information from other sequences. So a dropout layer is very important during training so as to generalize the global representation to all steps. In this way, we can distinguish the importance of different items for generating the global representation of the sequence, leading to better extraction of global intentions.

3.3 Disentangled Module

The local module and the global module can not completely fulfill our requirements of the sequence encoder. In particular, their ability at capturing multiple intentions is limited. The disentangled module is appended after local module and global module so as to reuse their expressive power.

3.3.1 Relevance weighting. According to the distance between the sequence embedding and a set of intention prototypes, we disentangle the user preference into k intentions. For brevity, we abbreviate LayerNorm as LN, which can be written as:

$$p_{k|i} = \frac{\exp\left(\frac{1}{\sqrt{d}}\text{LN}_1\left(\mathbf{e}_u^{(i)}\right)\cdot\text{LN}_2\left(\mathbf{c}_u^{(k)}\right)\right)}{\sum_{k'=1}^{K}\exp\left(\frac{1}{\sqrt{d}}\text{LN}_1\left(\mathbf{e}_u^{(i)}\right)\cdot\text{LN}_2\left(\mathbf{c}_u^{(k')}\right)\right)}, \tag{10}$$

where $p_{k|i}$ is to measure how likely the main intention at position i is related with the k-th latent intention, $\mathbf{e}_u^{(i)}$ is the global or local representation from the current user's historical behaviors, i.e., it is short for $Local(s_u^{(i)})$ or $Global(s_u^{(i)})$. $\mathbf{c}_u^{(k)} \in \mathbb{R}^d$ is the k-th prototype intention of users, $\text{LN}_l(\cdot)$ represents different layer normalization layers, which are distinguished by the l, since each layer normalization layer has its own parameters for scaling its output.

3.3.2 Attention weighting. In addition to the relevance weight $p_{k|i}$, we also consider attention weight p_i to measure how likely the intention at position i is essential for predicting the user's next intention.

$$\begin{aligned}
p_i &= \frac{\exp\left(\frac{1}{\sqrt{d}}\text{key}_i\cdot\text{ query}\right)}{\sum_{i'=1}^{L}\exp\left(\frac{1}{\sqrt{d}}\text{key}_{i'}\cdot\text{ query}\right)},\\
\text{query} &= \text{LN}_3\left(\varphi_t + \mathbf{e}_u^{(t)} + \rho\right),\\
\widehat{\text{key}}_i &= \text{LN}_4\left(\varphi_i + \mathbf{e}_u^{(i)}\right),\\
\text{key}_i &= \widehat{\text{key}}_i + \text{ReLU}\left(\mathbf{W}^\top\widehat{\text{key}}_i\right),
\end{aligned} \tag{11}$$

where $i = 1, 2, \ldots, t$, $\mathbf{W} \in \mathbb{R}^{d\times d}$, $\rho \in \mathbb{R}^d$, and $\varphi_i \in \mathbb{R}^d$ are learnable parameters. We introduce p_i to avoid focusing too much on the latest intention in the vector space, and the earlier intentions that are close to the latest intention in the vector space are more likely to be important. Since the above assumption may not always be correct, so it is necessary to introduce the learnable parameters φ_t and ρ.

3.3.3 Intention disentangling. We employ intention disentangling instead of intention aggregation, we disentangle the user's representation for each position i in a sequence $s_u^{(i)}$ under k latent categories according to $p_{k|i}$ and p_i. The k-th intention is computed as follows:

$$\begin{aligned}
LI(s_u^{(i)})^{(k)} &= \text{LN}_5\left(p_{k|i}\cdot p_i\cdot Local(s_u^{(i)})\right),\\
GI(s_u^{(i)})^{(k)} &= \text{LN}_5\left(p_{k|i}\cdot p_i\cdot Global(s_u^{(i)})\right),\\
Final\ Intentions(s_u^{(i)})^{(k)} &= LI(s_u^{(i)})^{(k)} + GI(s_u^{(i)})^{(k)}.
\end{aligned} \tag{12}$$

where $k \in \{1, 2, \ldots\}$, finally we add up the global intention $GI(s_u^{(i)})^{(k)}$ and the local intention $LI(s_u^{(i)})^{(k)}$ from the same intention prototypes k, thus each user gets k final intention representations for each position i.

3.4 Multi-Intention Contrastive Learning

3.4.1 Intent-Level Positive Pairs. Given a minibatch of N users $\{u_1, u_2, ..., u_N\}$, we apply two random augmentation operators a_i, a_j from set {C, M, R, S, I} for each user and obtain $2N$ augmented sequences $[s_{u_1}^{a_i}, s_{u_1}^{a_j}, s_{u_2}^{a_i}, s_{u_2}^{a_j}, ..., s_{u_N}^{a_i}, s_{u_N}^{a_j}]$. The multi-intention representations for augmented sequences can be viewed as follows:

$$\begin{aligned}
view1(s_u^{a_i})^{(k)} &= LI(s_u^{a_i})^{(k)} + GI(s_u^{a_i})^{(k)},\\
view2(s_u^{a_j})^{(k)} &= LI(s_u^{a_j})^{(k)} + GI(s_u^{a_j})^{(k)},
\end{aligned} \tag{13}$$

where $k \in \{1, 2, \ldots\}$, view1 and view2 means that we add the global intention representation and the local intention representation from the same intention prototype, thus we get k intent-level representations for each augmented sequence. As shown in Figure 2(d), since we generate two augmented sequences as inputs for two independent sequence encoders, we finally generate k intent-level positive pairs, i.e. $2 * k$ intention representations.

3.4.2 Contrastive Loss. In the CL task, our contrastive loss function is applied to distinguish whether the two representations are derived from the same intention in the same user historical sequence. With intention extraction from the sequence encoder, we can obtain $2N * k$ intent-level positive pairs in a minibatch of $\{u_1, u_2, ..., u_N\}$ users. We treat each pair $\left((s_u^{a_i})^{(k)}, (s_u^{a_j})^{(k)}\right)$ as a positive pair, and the other $2(N * k - 1)$ intent-level augmented sequences are considered as negative samples for this pair. Therefore, the loss function $\mathcal{L}_{\text{CL}}$ is optimized as follows:

$$\mathcal{L}_{\text{CL}}\left((s_u^{a_i})^{(k)}, (s_u^{a_j})^{(k)}\right) = -\log \frac{\exp\left(\text{sim}\left((s_u^{a_i})^{(k)}, (s_u^{a_j})^{(k)}\right)\right)}{\sum_{s \in neg} \exp\left(\text{sim}\left((s_u^{a_i})^{(k)}, s\right)\right)}. \tag{14}$$

where $\text{sim}(\cdot)$ is a dot product to measure the similarity between two shared intention representations. By creating high-quality views with intent-level, we can not only increase the number of negative samples to improve the performance of CL, but also keep the independence between different intentions. In this way, we can better extract the current main intention for the next-item prediction.

3.5 Multi-Task Training

In the SR task, we present next-item prediction in Figure 2(c). In a sequence, we match the main intention at the current position t for each positive and negative sample during the training stage, so as to select the main intention of the current user to predict the next item. Our SR loss for each user u is defined as follows:

$$\mathcal{L}_{\text{SR}}\left(s_u^t\right) = -\log \frac{\max_{k \in \{1,2,...,k\}} \left(\exp\left(s_u^{t\top} \cdot v_{t+1}^{+}\right)\right)}{\sum_{v'_{t+1} \in \mathcal{V}} \max_{k \in \{1,2,...,k\}} \left(\exp\left(s_u^{t\top} \cdot v'_{t+1}\right)\right)}, \tag{15}$$

To leverage the intent-level self-supervised signals derived from the unlabeled raw data to enhance the performance of sequential recommendation, we adopt a multi-task strategy where the SR task and the CL task are jointly optimized. The joint loss is a linear weighted sum calculated as:

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{SR}} + \lambda \mathcal{L}_{\text{CL}}. \tag{16}$$

where λ is a hyper-parameter controlling the weight of CL task loss.

4 EXPERIMENTS

In this section, we conduct extensive experiments to answer the following research questions:

- **RQ1:** Does IOCRec yield better recommendation?
- **RQ2:** How does multi-intention contrastive learning affect the performance of IOCRec?
- **RQ3:** How do different modules (e.g., global intention, etc.) affect the performance of IOCRec?
- **RQ4:** How do different hyper-parameters affect the performance of IOCRec?

4.1 Experimental Settings

4.1.1 Datasets. We conduct our experiments on four datasets collected from four real-world platforms with sparsity levels. The statistical details of all datasets after preprocessing are reported in Table 1, we briefly introduce their properties below.

Table 1: The statistics of the datasets.

Dataset	# Users	# Items	# Actions	Avg.length	Sparsity
Sports	25,598	18,357	296,337	8.3	99.95%
Beauty	22,363	12,101	198,502	8.9	99.73%
Yelp	30,431	20,033	316,354	10.3	99.95%
Toys	19,412	11,924	167,597	8.6	99.93%

(1) **Amazon Beauty, Sports, and Toys:** In this work, we select three subcategories from Amazon, they are obtained from Amazon review datasets in [26], which contain product reviews and abundant metadata.

(2) **Yelp**[1]**:** This is obtained from a business platform, note that we only use the transaction records after January 1st, 2019.

For all datasets, we group the interaction records by users and sort them by the interaction timestamps ascendingly. Following [22], we only keep the 5-core datasets, and discard users and items with fewer than 5 interactions. For each user, we adopt the last interacted item as the test data , the item just before the last as the validation data. The remaining items are used for training.

4.1.2 Evaluation Metrics. To conduct the performance evaluation, we employ two widely used metrics: Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG). We report results on HR@5, 10 and NDCG@5, 10. Since the item set is large, to avoid heavy computation on all user-item pairs, following the common strategy [15, 34, 48], we pair each ground-truth item in the test set with 99 randomly negative items that the user has not interacted with, and rank these items with the ground-truth item together.

4.1.3 Baseline Methods. We compare our proposed approach with the following nine baselines:

- **PopRec** is a simple method that ranks items according to their popularity measured by the number of associated actions.
- **GRU4Rec** [13] is a pioneering work that applies GRU to model user click sequence for session-based recommendation.
- **Caser** [33] is a CNN-based method capturing high-order Markov Chains for sequential recommendation.
- **BERT4Rec** [31] adopts a deep bidirectional self-attention model with mask mechanism in sequential recommendation.
- **SASRec** [15] uses a left-to-right transformer model with single-head attention mechanism to recommend the next item.
- **DSSRec** [25] utilizes seq2seq training and performs optimization in latent space for sequential recommendation.
- **S^3-Rec$_{\text{MIP,SP}}$** [48] uses SSL with a pre-training strategy to derive the intrinsic data correlation. In this section, we remove its attributes modules as we have no attributes for items, we only compare the mask item prediction (MIP) and sequence-segment correlation segment prediction (SP) in S^3-Rec for fairness.
- **CL4SRec** [43] fuses contrastive SSL with Transformer-based SR model, it only has crop, mask and reorder augmentation operators.
- **CoSeRec** [21] proposes two informative augmentation operators leveraging item correlations and fuses CL with Transformer based model.

[1] https://www.yelp.com/dataset

Table 2: Performance comparison of different methods on four datasets, where our approach IOCRec's best results are in bold. The underlined numbers are the best results besides IOCRec. The reported result of IOCRec for each dataset is the best result of applying intention k.

Dataset	Metric	PopRec	GRU4Rec	Caser	BERT4Rec	SASRec	DSSRec	S^3-Rec$_{MIP,SP}$	CL4SRec	CoSeRec	IOCRec$_{CL4S}$	IOCRec$_{CoSe}$	Improv.
Sports	NDCG@5	0.1538	0.2126	0.2020	0.2341	0.2497	0.2627	0.2594	0.2544	0.2543	**0.2885**	0.2856	9.82%
	NDCG@10	0.1902	0.2527	0.2390	0.2775	0.2869	0.2997	0.3035	0.2916	0.2927	**0.3272**	0.3249	7.81%
	HR@5	0.2293	0.3055	0.2866	0.3375	0.3466	0.3617	0.3673	0.3518	0.3510	**0.3950**	0.3915	7.54%
	HR@10	0.3423	0.4299	0.4014	0.4722	0.4622	0.4802	0.4933	0.4674	0.4699	**0.5152**	0.5130	4.44%
Beauty	NDCG@5	0.1391	0.2268	0.2219	0.2622	0.2848	0.2992	0.2657	0.2888	0.2887	**0.3215**	0.3202	7.45%
	NDCG@10	0.1803	0.2584	0.2512	0.2975	0.3156	0.3220	0.3018	0.3194	0.3202	**0.3535**	0.3511	9.78%
	HR@5	0.2105	0.3125	0.3032	0.3640	0.3741	0.3874	0.3682	0.3779	0.3774	**0.4166**	0.4153	7.54%
	HR@10	0.3386	0.4106	0.3942	0.4739	0.4696	0.4756	0.4805	0.4732	0.4751	**0.5161**	0.5112	7.41%
Yelp	NDCG@5	0.1622	0.3784	0.3696	0.4252	0.4113	0.4231	0.3634	0.4130	0.4183	**0.4662**	0.4659	9.64%
	NDCG@10	0.2007	0.4375	0.4198	0.4778	0.4642	0.4711	0.4268	0.4669	0.4718	0.5162	**0.5168**	8.16%
	HR@5	0.2415	0.5437	0.5111	0.5976	0.5745	0.5827	0.5256	0.5772	0.5836	0.6336	**0.6365**	6.51%
	HR@10	0.3609	0.7265	0.6661	0.7597	0.7373	0.7412	0.7233	0.7433	0.7483	0.7872	**0.7875**	3.66%
Toys	NDCG@5	0.1286	0.1919	0.1885	0.2327	0.2820	0.2934	0.2307	0.2859	0.2854	**0.3152**	0.3144	7.43%
	NDCG@10	0.1618	0.2274	0.2183	0.2698	0.3136	0.3256	0.2742	0.3173	0.3166	**0.3464**	0.3455	6.39%
	HR@5	0.1977	0.2795	0.2614	0.3344	0.3682	0.3723	0.3368	0.3749	0.3735	0.4071	**0.4078**	8.78%
	HR@10	0.3008	0.3896	0.3540	0.4493	0.4663	0.4798	0.4729	0.4723	0.4705	0.5032	**0.5041**	5.06%

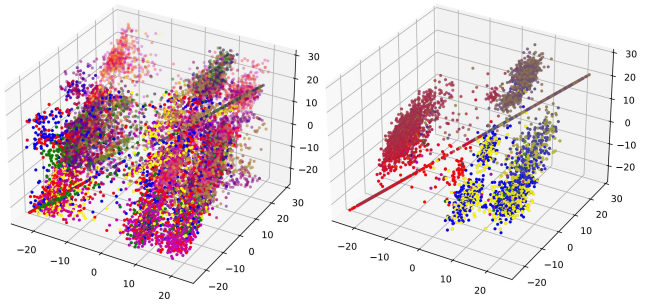
4.1.4 Implementation Details. For our baselines, all parameters are set following the suggestions from the original papers. Our method is implemented in PyTorch. According to CL4SRec [43], IOCRec$_{CL4S}$ randomly adopt {C,M,R} to augment all sequences, while IOCRec$_{CoSe}$ according to CoSeRec [21], we randomly adopt {S, I, M} for short sequences with less than 4 interactions and randomly select {S, I, M, R, C} to augment the long sequences. For our proposed IOCRec, we set 3 Transformer structures and 2 attention heads as local module. The dimension of the embedding is 64, the maximum sequence length is set to 50, the batch size and λ within {256, 1024} and {0.1, 0.2, ..., 0.5}. The number of latent intentions is tuned from {2, 3, ..., 6}. The model is optimized by Adam optimizer [16] with a learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and linear decay of the learning rate.

4.2 Performance Comparison (RQ1)

Table 2 presents the performance comparisons between several baselines and our model (IOCRec). Here, we can find: PopRec performs worse than sequential models in general, which indicates the importance of mining the sequential patterns.

As for sequential recommendation baseline methods, self attention mechanism models (SASRec and BERT4Rec) achieve better performance than traditional methods (Caser and GRU4Rec). DSSRec further improves SASRec's performance by using a seq2seq training strategy and reconstructing the representation of the future sequence in latent space. In addition, S^3-Rec adopts SSL to provide additional training signals to enhance representations, but we observe that the performance of S^3-Rec$_{MIP,SP}$ is inferior to SASRec in some datasets. Although the lack of extra attribute information is a potential influencing factor, another primary reason is that S^3-Rec$_{MIP,SP}$ adopts the two-stage training preventing information sharing between SSL and next-item prediction targets. After being augmented by CL, CL4SRec and CoSeRec consistently outperform SASRec slightly, which indicates the effectiveness of enhancing sequence representations via CL on an individual user level. We also find that CoSeRec exhibits worse performance than CL4SRec in some datasets. The reason might be that the insert and substitute augmentation operators in CL inevitably introduce noise, making it difficult to be broadly applicable to all datasets. The introduced noise also limits the effectiveness of CL.

Finally, IOCRec fuses multi-intention into SR model by a new CL, which helps the encoder discover a good semantic structure across user's behavior sequences. The IOCRec consistently outperforms existing methods on all datasets, showing the necessity of extracting multi-intention oriented CL for denoising problem.



(a) Multi-Intention without CL (b) Multi-Intention Oriented CL

Figure 3: Users' multi-intention representations comparing (a) and (b) on Yelp.

4.3 Multi-Intention Analysis (RQ2)

To evaluate the role of multi-intention contrastive learning, we randomly select 2000 users to visualize multi-intention representations. The visualization is based on PCA decomposition, which will project the embedding matrix into 3D. The results are shown in Figure 3, different colors represent different intentions. We disentangle multi-intention for each user on the Yelp dataset. In figure

Table 3: Ablation study of IOCRec (NDCG@10).

Model	Sports	Beauty	Yelp	Toys
$\text{IOCRec}_{\text{CL4S}}$	**0.3272**	**0.3535**	**0.5162**	**0.3464**
w/o GI	0.3196	0.3394	0.5150	0.3371
w/o LI	0.3135	0.3265	0.4856	0.3239
w/o IC	0.3130	0.3233	0.5043	0.3219
$\text{IOCRec}_{\text{CoSe}}$	**0.3249**	**0.3511**	**0.5168**	**0.3455**
w/o GI	0.3177	0.3433	0.5126	0.3363
w/o LI	0.3129	0.3251	0.4818	0.3221
w/o IC	0.3105	0.3291	0.4924	0.3235
CL4SRec	0.2916	0.3194	0.4669	0.3173
CL4SRec+IC	**0.3143**	**0.3311**	**0.4981**	**0.3318**
CoSeRec	0.2927	0.3202	0.4718	0.3166
CoSeRec+IC	**0.3168**	**0.3354**	**0.5017**	**0.3307**

3(a), we employ IOCRec's sequence encoder to generate multi-intention representations. Though multi-intention representations are effective. However, due to intentional diversity, it is still difficult to distinguish different intentions of users. While in figure 3(b), multi-intention oriented CL can separate different intentions, it will make representations from the same intention to be close, and those of dissimilar intentions to be far apart. Therefore, our new method IOCRec can better distinguish the main intention and other intentions for predicting the next item.

4.4 Ablation Study (RQ3)

In this section, we consider three variants for $\text{IOCRec}_{\text{CL4S}}$ and $\text{IOCRec}_{\text{CoSe}}$ respectively, the details are as follows:

- w/o GI: we only remove the global intention in the sequence encoder, so keep the local intention in multi-intention contrastive learning to create views.
- w/o LI: we only remove the local intention in the sequence encoder, so keep the global intention in multi-intention contrastive learning to create views.
- w/o IC: we only remove the contrastive learning task in IOCRec, and use the sequence encoder that integrate global and local intention representations to select the current main intention for recommendation.

Table 3 reports the performance (NDCG@10) comparison between IOCRec and its three variants. Moreover, we additionally add the disentangled module based on CL4SRec and CoSeRec respectively. When introducing disentangled module, we select the best intention k on the dataset respectively. We find that the CL4SRec+IC and CoSeRec+IC models also benefit from the contrastive learning of intention representations objective. For the IOCRec and its three variants, without local intentions, the performance of the IOCRec degrades significantly. This also shows that under the influence of only the global intentions, it is difficult to accurately capture the user's current intention. Meanwhile, in the absence of the contrastive learning task in IOCRec, although the sequence encoder that fuses global and local intentions can improve the recommendation performance, the effect is worse than CL4SRec+IC and CoSeRec+IC. This also just proves the importance of multi-intention oriented contrastive learning. Obviously, IOCRec obtains the best performance against these variants on four datasets.

4.5 Parameter Sensitivity (RQ4)

Due to the space limit, we only report the effect of some key hyper-parameters on the model performance.

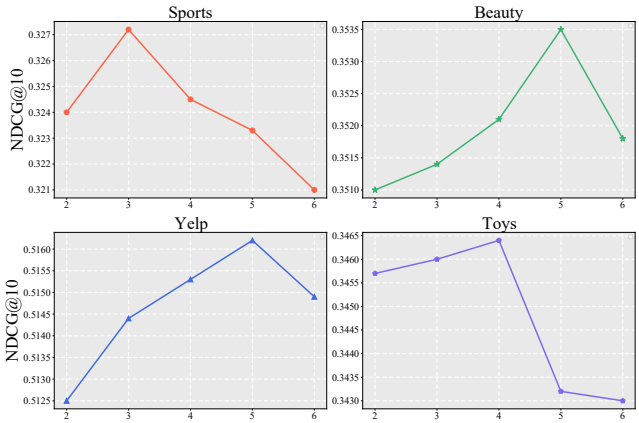


Figure 4: Performance (NDCG@10) comparison of the number k of intentions on four datasets.

Impact of the number k of intentions. It is crucial to find the optimal number k of intention on each dataset, which can subjectively reflect the real intention of users under each dataset. Figure 4 shows the impact of the number k of intentions on four public datasets. For the all datasets, IOCRec obtains the better performance when k = 3, 4, 5. Hence, setting too small and too big numbers of intentions cannot reflect the real situation of users.

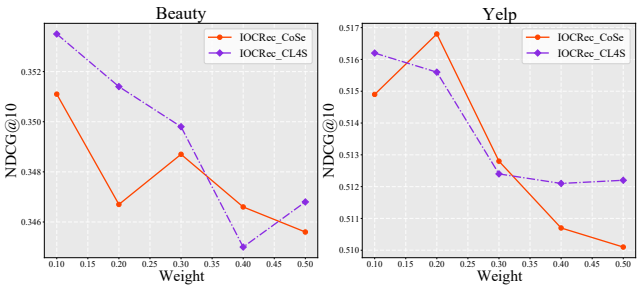


Figure 5: Performance (NDCG@10) comparison of the different λ.

Impact of contrastive learning loss. We investigate how the contrastive learning loss of our proposed IOCRec interacts with the sequential prediction loss. Specifically, we follow two data augmentation strategies, IOCRec and ICL4SRec, respectively. We select the best number k of intentions for each dataset and keep other parameters fixed to make a fair comparison. Figure 5 shows the

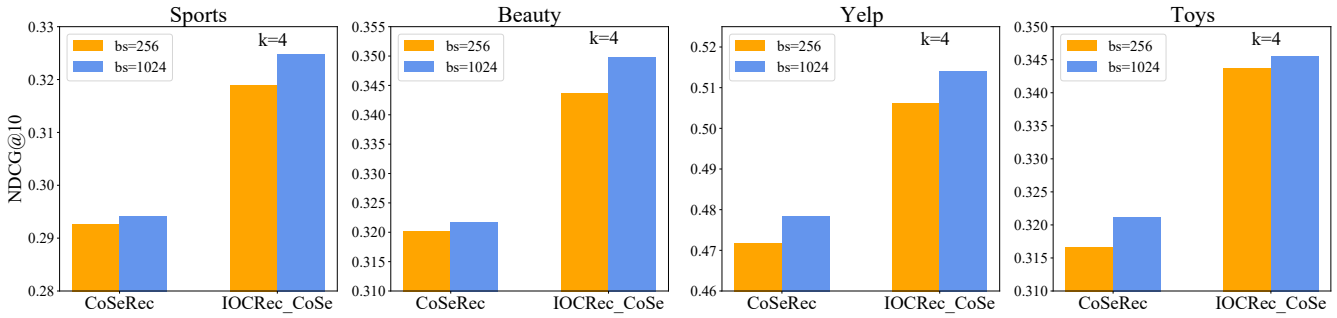


Figure 6: Performance (NDCG@10) comparison of different batch size on four datasets.

evaluation results. Note that with larger value λ, $\mathcal{L}_{\text{CL}}$ contributes more heavily in the $\mathcal{L}_{\text{joint}}$. In most cases, we observe that performance deteriorates when λ increases over than certain threshold (i.e., λ = 0.1).

Impact of negative sample size. As shown in Figure 6, during the training of CL, we demonstrate that increasing the batch size to capture user preferences is beneficial. We also additionally employ multi-intention representations to increase the negative samples. CoSeRec generates a pair of positive samples for each user, while we generate k pairs of positive samples for each user. Therefore, in CL, the number of negative samples is not only related to the batch size, but also related to the number of intention k. So we select a special case, where k = 4 and batch size = 256 in our work while the batch size = 1024 in our baselines to maintain the same number of negative samples to ensure fairness. We can observe that IOCRec deriving better representations, even if k=4 is not our best result on some datasets.

5 RELATED WORK

(1) Sequential Recommendation. Some earlier works [7] rely on Markov chains to capture correlation among items. Due to the nonlinear expressive capacity, LSTM [39] and GRU [13] are firstly introduced to the session-based recommendation. Furthermore, CNN-based models [33, 45] allow feature composition over the behaviors in the interaction sequence. Recently, the attention mechanism has expressed promising potential to capture context-aware preference for SR [2, 14, 15, 18, 31, 40]. However, ASReP [22] argues that these methods still limit their expression performance in sparse short sequences. Therefore, augmentation for short sequences is desirable.

(2) Self-supervised Learning. SSL has become an emerging trend in CV [3, 10, 12] and NLP areas [17]. The core of SSL is to explore effective information from unsupervised data, which can improve the quality of representation learning [8, 11, 28]. In the field of SR, S^3-Rec [48] adopts a pre-training and fine-tuning strategy, and first proposes using mutual information to maximize attributes and sequences. SGL [42] is a graph-based recommender system that adopts augmentations by node dropout, edge dropout, and random walk methods. RAP [35] proposes a pattern-enhanced contrastive policy learning network for denoising and recommendation. SSI [46] extracts the consistent knowledge by utilizing three SSL pre-training tasks. The recent work, CL4SRec [43] proposes three data augmentation techniques (i.e., cropping, masking and reordering) from which two views are randomly sampled and applied to each sequence. Based on that, CoSeRec [21] proposes two augmentation operators (substituting and inserting). However, in CL, two views obtained from the same user behavior sequence must be similar, even if they are formed from distinct intentions. Such one-size-fits-all approaches necessarily inject noise into representation learning, lowering CL performance.

(3) Disentangled Representation Learning. There are multiple potential intentions behind each user's behavior [5, 20, 32, 37, 41]. MacridVAE [24] is the first attempt that incorporate the disentangled representation learning into user historical data at both a macro and a micro level. Later, several GNN-based models [23, 38] have been proposed to disentangle the multiple intentions. In addition, DSSRec [25] proposes a sequence-to-sequence training strategy to extract extra supervision signals in the disentangled latent space. Instead, we disentangle the user intentions for a different purpose than above methods. Our purpose is to create high-quality views with intent-level, so as to guide the denoising process.

6 CONCLUSION

In this paper, we argue that data augmentation can amplify the noises in the original sequence, which is not suitable for SR. Based on this, we propose IOCRec to guide the denoising process. Firstly, we design a global module to capture users' global preferences, which is beneficial in fitting well with the local module. Secondly, we investigate an adaptive disentangled module to reuse global module and local module expressive power, then select the current main intention to improve the performance of SR. Finally, IOCRec enhances the role of CL in SR by considering the construction of positive pairs under the intent-level. Extensive experiments show that IOCRec achieves state-of-the-art performance against a series of SOTA solutions.

ACKNOWLEDGMENTS

This work is jointly supported by Tianjin Key Laboratory of Advanced Networking, Tianjin Key Laboratory of Cognitive Computing and Application, National Natural Science Foundation of China (61877043) and National Natural Science of China (61877044).

REFERENCES

[1] John S Breese, David Heckerman, and Carl Kadie. 2013. Empirical analysis of predictive algorithms for collaborative filtering. *arXiv preprint arXiv:1301.7363* (2013).

[2] Renqin Cai, Jibang Wu, Aidan San, Chong Wang, and Hongning Wang. 2021. Category-aware collaborative sequential recommendation. In *SIGIR*. 388–397.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*. PMLR, 1597–1607.

[4] Tong Chen, Hongzhi Yin, Quoc Viet Hung Nguyen, Wen-Chih Peng, Xue Li, and Xiaofang Zhou. 2020. Sequence-aware factorization machines for temporal predictive analytics. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 1405–1416.

[5] Wanyu Chen, Pengjie Ren, Fei Cai, Fei Sun, and Maarten de Rijke. 2020. Improving end-to-end sequential recommendations with intent-aware diversification. In *CIKM*. 175–184.

[6] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022. Intent Contrastive Learning for Sequential Recommendation. In *WWW*. 2172–2182.

[7] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. 2013. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI*. 2605–2611.

[8] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. 2020. Debiased Contrastive Learning. In *NeurIPS*.

[9] Ziwei Fan, Zhiwei Liu, Yu Wang, Alice Wang, Zahra Nazari, Lei Zheng, Hao Peng, and Philip S Yu. 2022. Sequential Recommendation via Stochastic Self-Attention. In *WWW*. 2036–2047.

[10] Tengda Han, Weidi Xie, and Andrew Zisserman. 2020. Memory-augmented dense predictive coding for video representation learning. In *ECCV*. Springer, 312–329.

[11] Tengda Han, Weidi Xie, and Andrew Zisserman. 2020. Self-supervised co-training for video representation learning. *Advances in Neural Information Processing Systems* 33 (2020), 5679–5690.

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*. 9729–9738.

[13] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *ICLR*.

[14] Wendi Ji, Keqiang Wang, Xiaoling Wang, Tingwei Chen, and Alexandra Cristea. 2020. Sequential recommender via time-aware attentive memory network. In *CIKM*. 565–574.

[15] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *ICDM*. 197–206.

[16] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[17] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *ICLR*.

[18] Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time interval aware self-attention for sequential recommendation. In *WSDM*. 322–330.

[19] Kaiyuan Li, Pengfei Wang, and Chenliang Li. 2022. Multi-Agent RL-based Information Selection Model for Sequential Recommendation. In *SIGIR*. 1622–1631.

[20] Zihan Lin, Hui Wang, Jingshu Mao, Wayne Xin Zhao, Cheng Wang, Peng Jiang, and Ji-Rong Wen. 2022. Feature-aware Diversified Re-ranking with Disentangled Representations for Relevant Recommendation. In *KDD*. 3327–3335.

[21] Zhiwei Liu, Yongjun Chen, Jia Li, Philip S Yu, Julian McAuley, and Caiming Xiong. 2021. Contrastive self-supervised sequential recommendation with robust augmentation. *arXiv preprint arXiv:2108.06479* (2021).

[22] Zhiwei Liu, Ziwei Fan, Yu Wang, and Philip S. Yu. 2021. Augmenting Sequential Recommendation with Pseudo-Prior Items via Reversely Pre-training Transformer. In *SIGIR*.

[23] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. 2019. Disentangled graph convolutional networks. In *ICML*. 4212–4221.

[24] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning Disentangled Representations for Recommendation. In *NeurIPS*. 5712–5723.

[25] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled self-supervision in sequential recommenders. In *KDD*. 483–491.

[26] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*. 43–52.

[27] Quoc Viet Hung Nguyen, Chi Thang Duong, Thanh Tam Nguyen, Matthias Weidlich, Karl Aberer, Hongzhi Yin, and Xiaofang Zhou. 2017. Argument discovery via crowdsourcing. *The VLDB Journal* 26, 4 (2017), 511–535.

[28] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *WSDM*. 813–823.

[29] Ruiyang Ren, Zhaoyang Liu, Yaliang Li, Wayne Xin Zhao, Hui Wang, Bolin Ding, and Ji-Rong Wen. 2020. Sequential recommendation with self-attentive multi-adversarial network. In *SIGIR*. 89–98.

[30] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *WWW*. 811–820.

[31] 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 *CIKM*. 1441–1450.

[32] Qiaoyu Tan, Jianwei Zhang, Jiangchao Yao, Ninghao Liu, Jingren Zhou, Hongxia Yang, and Xia Hu. 2021. Sparse-interest network for sequential recommendation. In *WSDM*. 598–606.

[33] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*. 565–573.

[34] Md Mehrab Tanjim, Congzhe Su, Ethan Benjamin, Diane Hu, Liangjie Hong, and Julian McAuley. 2020. Attentive sequential models of latent intent for next item recommendation. In *WWW*. 2528–2534.

[35] Xiaohai Tong, Pengfei Wang, Chenliang Li, Long Xia, and Shaozhang Niu. 2021. Pattern-enhanced Contrastive Policy Learning Network for Sequential Recommendation. In *IJCAI*. 1593–1599.

[36] 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 *NIPS*. 5998–6008.

[37] 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 *IJCAI*.

[38] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *SIGIR*. 1001–1010.

[39] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J. Smola, and How Jing. 2017. Recurrent Recommender Networks. In *WSDM*. 495–503.

[40] Jibang Wu, Renqin Cai, and Hongning Wang. 2020. Déjà vu: A contextualized temporal attention mechanism for sequential recommendation. In *WWW*. 2199–2209.

[41] Jiahao Wu, Wenqi Fan, Jingfan Chen, Shengcai Liu, Qing Li, and Ke Tang. 2022. Disentangled contrastive learning for social recommendation. In *CIKM*. 4570–4574.

[42] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *SIGIR*. 726–735.

[43] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Bolin Ding, and Bin Cui. 2020. Contrastive Learning for Sequential Recommendation. *arXiv preprint arXiv:2010.14395* (2020).

[44] Yuhao Yang, Chao Huang, Lianghao Xia, Yuxuan Liang, Yanwei Yu, and Chenliang Li. 2022. Multi-Behavior Hypergraph-Enhanced Transformer for Sequential Recommendation. In *KDD*. 2263–2274.

[45] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. 2019. A simple convolutional generative network for next item recommendation. In *WSDM*. 582–590.

[46] Xu Yuan, Hongshen Chen, Yonghao Song, Xiaofang Zhao, Zhuoye Ding, Zhen He, and Bo Long. 2021. Improving Sequential Recommendation Consistency with Self-Supervised Imitation. In *IJCAI*. 3321–3327.

[47] Yan Zhang, Hongzhi Yin, Zi Huang, Xingzhong Du, Guowu Yang, and Defu Lian. 2018. Discrete deep learning for fast content-aware recommendation. In *WSDM*. 717–726.

[48] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *CIKM*. 1893–1902.