Intent Contrastive Learning Based on Multi-view Augmentation for Sequential Recommendation

Bo Pei[1], Yingzheng Zhu[1], Guangjin Wang[1], Huajuan Duan[2], Wenya Wu[1], Fuyong Xu[1], Yizhao Zhu[1], Peiyu Liu[1]†, Ran Lu[1]†
p15264026338@163.com,liupy@sdnu.edu.cn,luran@sdnu.edu.cn
[1]School of Information Science and Engineering, Shandong Normal University
[2]School of Information Engineering, Shandong Management University

Abstract

Sequential recommendation systems play a key role in modern information retrieval. However, existing intent-related work fails to adequately capture long-term dependencies in user behavior, i.e., the influence of early user behavior on current behavior, and also fails to effectively utilize item relevance. To this end, we propose a novel sequential recommendation framework to overcome the above limitations, called ICMA. Specifically, we combine temporal variability with position encoding that has extrapolation properties to encode sequences, thereby expanding the model's view of user behavior and capturing long-term user dependencies more effectively. Additionally, we design a multi-view data augmentation method, i.e., based on random data augmentation methods (e.g., crop, mask, and reorder), and further introduce insertion and substitution operations to augment the sequence data from different views by utilizing item relevance. Within this framework, clustering is performed to learn intent distributions, and these learned intents are integrated into the sequential recommendation model via contrastive SSL, which maximizes consistency between sequence views and their corresponding intents. The training process alternates between the Expectation (E) step and the Maximization (M) step. Experiments on three real datasets show that our approach improves by 0.8% to 14.7% compared to most baselines.

1 Introduction

Recommendation systems are widely applied in various scenarios to accurately predict users' interests in a large number of items based on their historical interactions. With the development of recommendation system, sequential recommendation (Qin et al., 2023; Fan et al., 2023; Wang et al., 2024) has gradually become a research hotspot, and the dynamic interest of users can be captured more accurately by analyzing user interaction sequence (Zhu et al., 2024; Lee et al., 2023; Fan et al., 2021).

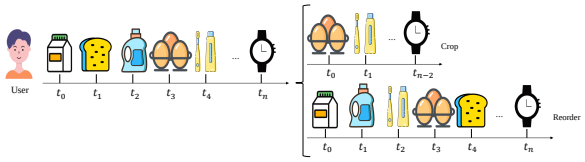


Figure 1: Examples of crop and reorder, they fail to take full advantage of the relevance between items.

Users' purchasing behavior is largely influenced by intent. However, the accurate use of intent has not been fully explored. Most existing user intent modeling methods (Cai et al., 2021; Zhang et al., 2023) rely on auxiliary information. For example, CoCoRec (Cai et al., 2021) makes use of item category information, but categorical features alone are not sufficient to accurately represent user intent, e.g., an intent such as "buy clothes" may relate to several different categories of items rather than being limited to a single category. DSSRec (Ma et al., 2020) proposes a seq2seq training strategy to optimize intent in potential space. However, DSSRec only deduces intents based on individual sequence representations, ignoring the potential correlation of different user intents.

Although the above methods achieve some success in capturing user intent, they typically treat items in a user's sequence as independent entities, ignoring the relevance between them. Fig 1 shows the items a user purchases over a month. Random augmentation methods like cropping or reordering treat these items as independent, potentially overlooking their relevance. For instance, cropping may retain only eggs and toothpaste, losing relevance information, while reordering may disrupt the sequence, such as bread, laundry detergent, and eggs. However, these items may be relevant, for example, milk, bread, and eggs are usually breakfast com-

†Corresponding author

Proceedings of the 31st International Conference on Computational Linguistics, pages 3300–3309
January 19–24, 2025.

ponents, and laundry detergent and toothpaste are household essentials. Additionally, early behaviors may influence user intent, such as repeatedly purchasing electronics, which indicates a continuing interest in those items. Therefore, capturing long-term dependencies is crucial for modeling user intent, considering the impact of early behaviors.

To address the above problems, we propose ICMA. Specifically, unlike traditional positional embedding, we use extrapolated position encoding, which enables the model to better handle sequences of different lengths. This is accomplished by extrapolating the encoded positions, which is essential to capture the long-term dependencies of users. To effectively utilize item relevance, we design a multi-view data augmentation method, which further introduces two kinds of augmentation operators, insertion and substitution, based on the random data augmentation method. The insertion operator adds relevant items to sequences to simulate the expansion of users' interests and enhance the model's ability to adapt to the new items, and the substitution operator replaces the relevant items to strengthen the model's ability to sense the changes in users' interests. These two operators not only expand the user interaction record, but also better capture the complex intentions and dynamic interests of users through diversified interaction modes. Sequences are clustered in the framework and the learned intents are applied to the SR model by comparing Self-Supervised Learning (SSL) to maximize the consistency between sequence views and their corresponding intents. The main contributions of this paper can be summarized as follows:

- We propose ICMA, which makes a wider field of view available to the model by extrapolating the position encoding, enhancing the understanding of the user's intention.
- We design a multi-view data augmentation method to enhance the model's utilization of item relevance by introducing two operators.
- Extensive experiments on three datasets validate the effectiveness of our method, with performance improvements ranging from 0.8% to 14.7% compared to most baselines.

2 Related Work

2.1 Sequential Recommendation

Sequential recommendation predicts users' future interests based on historical behavior data (Chen et al., 2022a; Li et al., 2023a). Early work used Markov chains (Rendle, 2010; He and McAuley, 2016) to model item transition relationships. The success of Transformer (Vaswani, 2017) has driven the development of SR models based on it, such as SASRec (Kang and McAuley, 2018), which uses Transformer layers to learn item dependencies in sequences. Research shows that existing models perform poorly on short sequences (Liu et al., 2021b), making short sequence augmentation necessary. S^3 -Rec (Zhou et al., 2020) and CLS4Rec (Xie et al., 2022) explore contrastive learning with weak self-supervision signals but fail to address item relevance, and their performance in capturing long-term dependencies is limited. In contrast, ICMA introduces insertion and substitution operations to leverage item correlation by adding related items to the sequence, increasing the number of user interactions, enhancing the model's ability to capture users' dynamic interests, and using extrapolated position encoding to effectively capture long-term user dependencies.

2.2 Contrastive Self-Supervised Learning

Contrastive self-supervised learning has shown remarkable success in computer vision (CV) (He et al., 2020; Du et al., 2023), natural language processing (NLP) (Gunel et al., 2020; Du et al., 2024), and recommendation (Zhou et al., 2020; Xie et al., 2022). The basic goal of contrast SSL is to learn useful feature representations by bringing different enhanced views of the same data close together in the presentation space and keeping representations of different data apart. CL4SRec (Xie et al., 2022) uses a multi-task training framework with contrasting objectives to enhance user representations, but it adopts a random augmentation method and ignores the item relevance in the sequence.

Inspired by the above methods, ICMA utilizes contrastive learning and is aware of the key factor of user intent, which enables it to more accurately predict users' future interests and needs.

3 Methodology

In this section, we introduce ICMA, the overall framework is shown in Fig 2. First, we introduce extrapolated positional encoding in the embedding layer, which differs from traditional positional encoding and effectively captures users' long-term dependencies. Next, we design a multi-view data augmentation method that introduces two augmen-

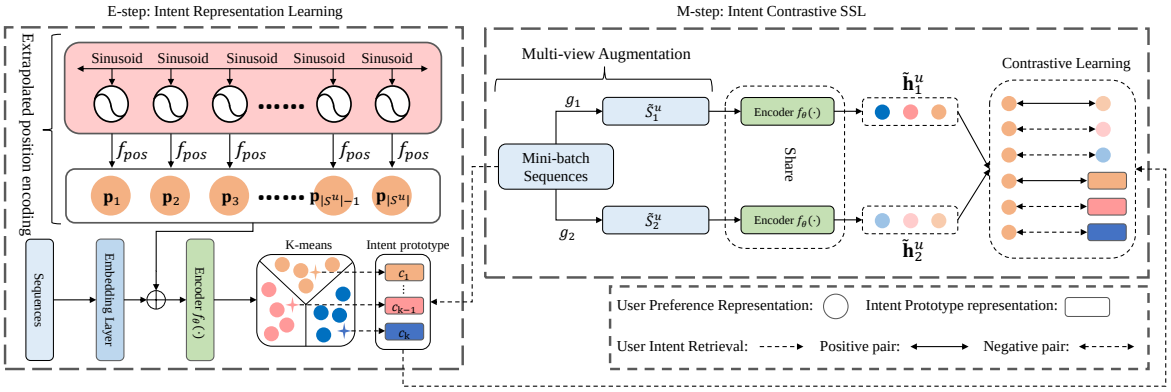


Figure 2: Overall Architecture of ICMA.

tation operators to enrich users' interaction records by utilizing item relevance. These improvements enable a more accurate reflection of user intent.

3.1 Problem Definition

Assume that the set of users and the set of items are denoted as $\mathcal{U}$ and $\mathcal{V}$, where $u \in \mathcal{U}$, $v \in \mathcal{V}$. For each user u, there is a sequence $S^u = \left[v_1, \cdots, v_t, \cdots, v_{|S^u|}\right]$ in chronological order, where $v_t \in \mathcal{V}$ denotes the interacting item at position t of user u in the sequence, and —S^u— denotes the total number of items. Represent $\mathbf{S}^u$ as the embedding representation of S^u and $\mathbf{v}_t$ is the d-dimensional embedding of the item v_t. In practice, the sequence is truncated to the maximum length T. If the sequence length is greater than T, the latest T items in the sequence are considered. If the sequence length is less than T, add a padding item at the beginning of the sequence until the length is T. Given the sequence S^u, the goal of SR is to recommend the items in the set $\mathcal{V}$ that the user u might interact with in the —S^u—+1 step.

3.2 Embedding Layer

The embedding layer contains an item embedding layer and a position embedding layer. The item embedding layer maps each item to a high-dimensional vector space to capture the semantic information and features of the item. The position embedding layer, on the other hand, captures the positional information of each item in the sequence to help the model understand the relative positions and order relationships of the items.

Extrapolated position encoding Conventional positional encoding methods perform well when dealing with fixed-length sequences, but are often overwhelmed when dealing with long sequences and are unable to accurately represent positional information beyond the encoding range. In order to overcome the limitations of traditional positional encoding, we propose a new encoding method, i.e., extrapolated position encoding, which provides more information dimensions and a rich representation of positional information by expanding the frequencies of the sine and cosine functions, so that the positional encoding is still expressive in a wider range, and thus is able to efficiently capture the user's long-term dependency relationships. The specific formula(Vaswani, 2017) is:

$$\begin{aligned} \mathrm{PE}(\mathrm{pos}, 2i) &= \sin\left(\frac{\mathrm{pos}}{10000^{2i/d_{\mathrm{model}}}}\right) \\ \mathrm{PE}(\mathrm{pos}, 2i+1) &= \cos\left(\frac{\mathrm{pos}}{10000^{2i/d_{\mathrm{model}}}}\right), \end{aligned} \tag{1}$$

where pos is the position and i is the dimension. For any fixed offset k, $\mathrm{PE}_{(\mathrm{pos+k})}$ can be represented as a linear function of $\mathrm{PE}_{\mathrm{pos}}$. $\mathrm{PE}_{\mathrm{pos}}$ is the position encoding vector for position pos. Based on the above equation we construct a position encoding matrix $E_p = \left\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{|S^u|}\right\}$, where $E_p \in \mathbb{R}^{|S^u| \times d}$.

Item embedding We map each item v_t in the input sequence to an embedding vector $\mathbf{v}_t$, forming the item embedding matrix $V = \left\{\mathbf{v}_1, \ldots, \mathbf{v}_t, \ldots \mathbf{v}_{|S^u|}\right\}$ of the user's history sequence, where $V \in \mathbb{R}^{|S^u| \times d}$. The initial vector is obtained by summing the item embedding and the item position encoding:

$$\mathbf{a}_{|S^u|} = \mathbf{v}_{|S^u|} + \mathbf{p}_{|S^u|}, \tag{2}$$

the initial input matrix corresponding to the item sequence is $\mathbf{S}^u = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_{|S^u|}]$, and the position encoding and item embedding are output separately to avoid noise interference.

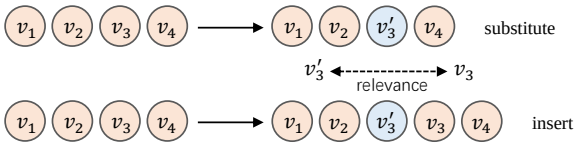


Figure 3: Substitution and insertion combine relevance between items

3.3 Multi-view Data Augmentation

Multi-view data augmentation is based on random data augmentation methods (e.g., crop, mask, reorder) and further introduces insertion and substitution operators, aiming to exploit item relevance more efficiently, to enrich the user's record of interactions, and to model the expansion of user interests.

3.3.1 Insertion and substitution

We introduce substitution and insertion operations to enhance data using item relevance. Figure 3 illustrates these two operations with a sequence $S^u = [v_1, v_2, \ldots, v_t]$, where $t = 4$. In the substitution operation, item v_3 is replaced by its relevant item v'_3, maintaining the original sequence order while enhancing data diversity and the model's understanding of item relevance. In the insertion operation, the relevant item v'_3 is inserted between v_2 and v_3, extending the user interaction sequence and improving the model's ability to capture item relevance.

Substitute(S). Randomly choose k different indexes $\{x_1, x_2, \ldots, x_k\}$ in the sequence S^u, where $k = \lceil \alpha t \rceil$ and each index x_i satisfies $x_i \in [1, 2, \ldots, t]$. $\alpha \in [0, 1]$ is the substitution rate. Where v'_{x_i} is the item relevant to v_{x_i}. Replace each item in these indices with the relevant item. The order of substitution is:

$$S^u_{\mathrm{S}} = \left[v_1, v_2, \ldots, v'_{x_i}, \ldots, v_t\right]. \tag{3}$$

Insert(I). The positions in the sequence S^u are randomly chosen to insert items, and the number of inserted items is controlled by the ratio $\omega \in [0, 1]$. We first choose k different index $\{x_1, x_2, \ldots, x_k\}$ in the sequence S^u, where $k = \lceil \omega t \rceil$ and each index x_i satisfies $x_i \in [1, 2, \ldots, t]$. Where v'_{x_i} is the most relevant item to v_{x_i}. We insert the relevant items at these indices. The order after insertion is:

$$S^{\mathrm{I}}_{\mathrm{S}} = \left[v_1, v_2, \ldots, v'_{x_i}, v_{x_i}, \ldots, v_t\right]. \tag{4}$$

3.3.2 Augmentation based on sequence length

Substitution and insertion are used for data augmentation based on item relevance. There are two ways to calculate item relevance, one is to calculate item relevance based on collaborative filtering of items (ItemCFIUF (Breese et al., 2013)). The relevance scores of items v_{x_i} and v'_{x_i} are defined as:

$$\mathrm{Rel}_{\mathrm{o}}(v_{x_i}, v'_{x_i}) = \frac{1}{\sqrt{|\mathcal{N}(v_{x_i})| \cdot |\mathcal{N}(v'_{x_i})|}} \cdot \sum_{u \in \mathcal{N}(v_{x_i}) \cap \mathcal{N}(v'_{x_i})} \frac{1}{\log\left(1 + |\mathcal{N}(u)|\right)}, \tag{5}$$

where u represents the user, $\mathcal{N}(v_{x_i})$ and $\mathcal{N}(v'_{x_i})$ denote the number of users who have interacted with items v_{x_i} and v'_{x_i}, respectively. Another uses dot product as a similarity measure. Given the representations of items v_{x_i} and v'_{x_i} as $e_{v_{x_i}}$ and $e_{v'_{x_i}}$, the relevance score is defined as:

$$\mathrm{Rel}_e(v_{x_i}, v'_{x_i}) = \mathrm{e}_{v_{x_i}} \cdot \mathrm{e}_{v'_{x_i}}. \tag{6}$$

We combine these two methods and take the highest value of both methods to calculate the relevance score(Liu et al., 2021a):

$$\mathrm{Rel}_{\mathrm{h}}(v_{x_i}, v'_{x_i}) = \max(\overline{\mathrm{Rel}}_{\mathrm{o}}(v_{x_i}, v'_{x_i}), \overline{\mathrm{Rel}}_{\mathrm{e}}(v_{x_i}, v'_{x_i})), \tag{7}$$

where $\overline{\mathrm{Rel}}_{\mathrm{o}}$ and $\overline{\mathrm{Rel}}_{\mathrm{e}}$ are the normalized scores of the above two methods, respectively.

Since short sequences may result in sparse data or insufficient information, we use different sets of augmentation operators for sequences depending on their length. The hyperparameter W determines whether the sequence is short or long and then the data augmentation is applied as follows:

$$S^u_a = \begin{cases} a(S^u), a \sim \{\mathrm{S}, \mathrm{I}, \mathrm{M}\}, & |S^u| \leq W \\ a(S^u), a \sim \{\mathrm{S}, \mathrm{I}, \mathrm{M}, \mathrm{C}, \mathrm{R}\}, & |S^u| > W \end{cases} \tag{8}$$

where a is the augmentation operator selected from the corresponding augmentation set. Although M produces fewer items when dealing with short sequences, we include it in the augmentation set of short sequences because it is able to preserve and reveal more complex and deeper relationships within the sequences to some extent.

3.4 Intent Representation Learning

We represent the Transformer encoder as $f_\theta(\cdot)$, and apply extrapolated position encoding to the sequence to capture the positional information of each item in the sequence. Subsequently, the sequence embedding $\mathbf{S}^u$ is encoded to output the user interest representation on all positional steps as:

$$\mathbf{h}^u = f_\theta(\mathbf{S}^u). \tag{9}$$

By maximizing the log-likelihood function $L(\theta)$, the optimal model parameter θ can be found:

$$L(\theta) = \sum_{u=1}^{N} \sum_{t=1}^{T} \ln P_\theta(v_t), \tag{10}$$

where P_θ represents the probability distribution function with parameter θ. This is equivalent to minimizing the prediction loss for sequential recommendation using the cross-entropy function:

$$\begin{aligned} \mathcal{L}_{Next} = & \sum_{u=1}^{N} \sum_{t=1}^{T} -\log\left(\sigma\left(\mathbf{h}_{t-1}^{u} \cdot \mathbf{v}_t\right)\right) \\ & - \sum_{\text{neg}} \log\left(1 - \sigma\left(\mathbf{h}_{t-1}^{u} \cdot \mathbf{v}_{\text{neg}}\right)\right), \end{aligned} \tag{11}$$

$\mathbf{h}_{t-1}^{u}$ denotes the user's interest in the position $t-1$, where $\mathbf{v}_t$ and $\mathbf{v}_{neg}$ denote embedding of the target term v_t and all items not interacted with by u. σ is a nonlinear activation function. N refers to the number of sequences in a small batch. Suppose there are K different user intentions, forming the intention variable $c = \{c_i\}_{i=1}^{K}$, the probability that a user interacts with an item is:

$$P_\theta(v) = \sum_{i=1}^{K} P_\theta(v \mid c_i) P(c_i). \tag{12}$$

Since user intent is latent, we cannot know the value of the variable c_i directly. Without c_i, the current parameter θ cannot be estimated, and without θ, there is no way to infer what the value of c_i might be. We use EM algorithm to solve the above problem. First guessing the value of θ and estimating the value of the missing variable c_i is the E-step. The M-step is to maximize the expected log-likelihood function and update the model parameter θ after obtaining the value of c_i. This process repeats until the log-likelihood converges.

Suppose there are K latent intent prototypes $\{c_i\}_{i=1}^{K}$ influencing the user to interact with the item, then based on Eq.(10) and Eq.(12), we can reformulate the maximum likelihood function as:

$$L(\theta) = \sum_{u=1}^{N} \sum_{t=1}^{T} \ln\left(\sum_{i=1}^{K} P_\theta(v_t \mid c_i) P(c_i)\right). \tag{13}$$

To facilitate the optimization of Eq.(13), we introduce the distribution $Q(c_i)$ of the hidden variable and rerepresent the sum within the logarithm:

$$\begin{aligned} L(\theta) &= \sum_{u=1}^{N} \sum_{t=1}^{T} \ln\left(\sum_{i=1}^{K} P_\theta(v_t, c_i)\right) \\ &= \sum_{u=1}^{N} \sum_{t=1}^{T} \ln \sum_{i=1}^{K} Q(c_i) \frac{P_\theta(v_t, c_i)}{Q(c_i)}. \end{aligned} \tag{14}$$

Based on Jensen's inequality, Eq.(14) is

$$\begin{aligned} &\geq \sum_{u=1}^{N} \sum_{t=1}^{T} \sum_{i=1}^{K} Q(c_i) \ln \frac{P_\theta(v_t, c_i)}{Q(c_i)} \\ &\geq \sum_{u=1}^{N} \sum_{i=1}^{K} Q(c_i) \left(\ln P_\theta(S^u, c_i) - \ln Q(c_i)\right). \end{aligned} \tag{15}$$

Here, we use $P_\theta(S^u, c_i)$ for $\prod_{t=1}^{T} P_\theta(v_t \mid c_i)$. In the optimization process, we only care about the terms related to the parameter θ. Therefore, removing the terms associated with terms not related to θ, the final lower bound function takes the form:

$$\sum_{u=1}^{N} \sum_{i=1}^{K} Q(c_i) \cdot \ln P_\theta(S^u, c_i), \tag{16}$$

where $Q(c_i) = P_\theta(c_i|S^u)$. Since $Q(c)$ is unknown, we cannot optimize Eq.(16) directly. Therefore, we use an alternating optimization method between the E-step and the M-step. In order to learn the user's intent distribution function $Q(c)$, we perform K-means (Chen et al., 2022b) clustering on all sequences. Following this, the distribution function $Q(c_i)$ is defined as:

$$Q(c_i) = P_\theta(c_i|S^u) = \begin{cases} 1, & \text{if } S^u \text{ in cluster } i \\ 0, & \text{else.} \end{cases} \tag{17}$$

3.5 Intent Contrastive SSL

We utilize contrastive SSL to fuse correlations between different views of a sequence. With the multi-view data augmentation method, given the sequence S^u, two augmentation views can be created (Zhou et al., 2020; Xie et al., 2022):

$$\tilde{S}_1^u = g_1(S^u), \tilde{S}_2^u = g_2(S^u), s.t. g_1, g_2 \sim \mathcal{G}, \tag{18}$$

where $\mathcal{G}$ is the set of predefined data augmentation functions, and g_1 and g_2 represent augmentation functions sampled from $\mathcal{G}$ to create different views for S^u. Typically, views created from the same sequence are treated as positive pairs, and any views from different sequences are treated as negative pairs. Enhanced views are first processed using extrapolated position encoding and then encoded as vector representations $\widetilde{\mathbf{h}}_1^u$ and $\widetilde{\mathbf{h}}_2^u$ by Transformer encoder $f_\theta(\cdot)$. We denote the contrastive loss as:

$$\begin{aligned} \mathcal{L}_{SCL} = & -\log \frac{\exp(\text{sim}(\tilde{\mathbf{h}}_1^u, \tilde{\mathbf{h}}_2^u))}{\sum_{neg} \exp(\text{sim}(\tilde{\mathbf{h}}_1^u, \tilde{\mathbf{h}}_{neg}))} \\ & -\log \frac{\exp(\text{sim}(\tilde{\mathbf{h}}_2^u, \tilde{\mathbf{h}}_1^u))}{\sum_{neg} \exp(\text{sim}(\tilde{\mathbf{h}}_2^u, \tilde{\mathbf{h}}_{neg}))}, \end{aligned} \tag{19}$$

where sim(·) is the dot product, and $\tilde{\mathbf{h}}_{neg}$ is the negative view representation of the sequence S^u. We have estimated the intent distribution function $Q(c)$, and to maximize Eq.(16), we borrow the formula proposed by Chen (Chen et al., 2022b) in order to redefine $P_\theta\left(S^u, c_i\right)$ as follows:

$$\begin{aligned} P_\theta(S^u, c_i) =& \frac{1}{K} \cdot P_\theta\left(S^u \mid c_i\right) \\ \propto & \frac{1}{K} \cdot \frac{\exp\left(\mathbf{h}^u \cdot \mathbf{c}_i\right)}{\sum_{j=1}^{K} \exp\left(\mathbf{h}^u \cdot \mathbf{c}_j\right)}. \end{aligned} \tag{20}$$

where $\mathbf{h}^u$ and $\mathbf{c}_u$ are vector representations of S^u and c_i, respectively. Based Eq.(16),(17),(20), maximizing Eq.(16) is equivalent to minimizing the following loss function:

$$-\sum_{v=1}^{N} \log \frac{\exp(\operatorname{sim}(\mathbf{h}^u, \mathbf{c}_i))}{\sum_{j=1}^{K} \exp(\operatorname{sim}(\mathbf{h}^u, \mathbf{c}_j))}, \tag{21}$$

this loss function is quoted from (Chen et al., 2022b), where sim(·) is the dot product. Given a batch of training sequences $\{s_u\}_{u=1}^{N}$, we first create two positive views of the sequences through Eq.(18) and then to optimize the following loss function:

$$\begin{aligned} \mathcal{L}_{IntCL} = & -\log \frac{\exp(\operatorname{sim}(\tilde{\mathbf{h}}_1^u, \mathbf{c}_u))}{\sum_{neg} \exp(\operatorname{sim}(\tilde{\mathbf{h}}_1^u, \mathbf{c}_{neg}))} \\ & -\log \frac{\exp(\operatorname{sim}(\tilde{\mathbf{h}}_2^u, \mathbf{c}_u))}{\sum_{neg} \exp(\operatorname{sim}(\tilde{\mathbf{h}}_2^u, \mathbf{c}_{neg}))}. \end{aligned} \tag{22}$$

where $\mathbf{c}_{neg}$ are all the intents in the given batch.

Datasets	Beauty	Sports	Toys
#Users	22,363	35,598	19,412
#Items	12,101	18,357	11,924
#Actions	198,502	296,337	167,597
Avg.length	8.9	8.3	8.6
Sparsity	99.95%	99.95%	99.93%

Table 1: Statistics of the experimented datasets.

3.6 Joint loss

We use joint loss to train the SR model, where the main next-item prediction, IntCL, and contrastive SSL are jointly optimized. It is shown as follows:

$$\mathcal{L} = \mathcal{L}_{Next} + \lambda \cdot \mathcal{L}_{IntCL} + \beta \cdot \mathcal{L}_{SCL}, \tag{23}$$

where λ and β control the intensity of the IntCL task and sequence-level SSL task, respectively.

4 Experiments

In this section, we conduct experiments on three datasets to evaluate our model ICMA and remove modules from the model to conduct experiments to verify the validity of the modules and also analyze the effect of hyperparameters.

4.1 Datasets

We conduct experiments on three public data sets. Sports, Beauty and Toys are the three subcategories of Amazon review data introduced in (McAuley et al., 2015). We follow (Xie et al., 2022) to prepare the datasets, and only keep the "5 core" dataset where all users and items have at least 5 interactions. Table **??** lists the detailed statistics for the three datasets.

4.2 Evaluation Metrics and parameter setup

We rank the entire item set and evaluate performance using HR and NDCG. We optimize using the Adam optimizer. The learning rate is 0.001. Set the self-attention block and attention head to 2, and the dimension of the embedding as 128. The batch size is 256.

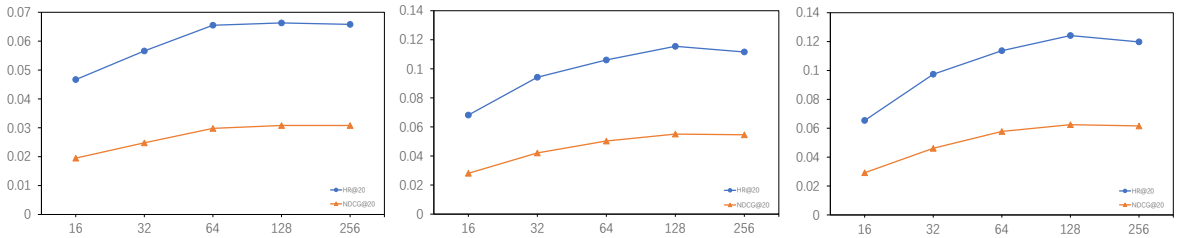
4.3 Baseline Models

We compare ICMA with the following representative SR models:

- **SASRec** (Kang and McAuley, 2018): This model utilizes an attention network for recommendation, which greatly improves the performance of SR.
- **BERT4Rec** (Sun et al., 2019): This model replaces the next prediction with a completion task to fuse information between items (views) in a sequence of user behaviors and their contextual information.
- **$\mathbf{S^3}$-Rec** (Zhou et al., 2020): It introduces SSL to capture correlations between items in a given sequence.
- **CL4SRec** (Xie et al., 2022): The model combines data augmentation with contrastive SSL using a random data augmentation method.
- **DSSRec** (Ma et al., 2020): It introduces the seq2seq training strategy and the intent unwrapping layer for SR.

Dataset	Metric	SASRec	DSSRec	BERT4Rec	S^3-Rec	CL4SRec	ICLRec	IOCRec	S^4Rec	ICMA	impr.
Sports	HR@5	0.0206	0.0214	0.0217	0.0121	0.0217	0.0290	0.0284	0.0293	**0.0306**	4.4%
	HR@20	0.0497	0.0495	0.0604	0.0344	0.0540	0.0646	**0.0684**	0.0656	0.0663	-
	NDCG@5	0.0135	0.0142	0.0143	0.0084	0.0137	0.0191	0.0169	0.0181	**0.0208**	8.9%
	NDCG@20	0.0216	0.0220	0.0251	0.0146	0.0227	0.0291	0.0279	0.0292	**0.0308**	5.5%
Beauty	HR@5	0.0374	0.0410	0.0360	0.0189	0.0423	0.0500	0.0511	0.0519	**0.0568**	9.4%
	HR@20	0.0901	0.0914	0.0984	0.0487	0.0994	0.1058	0.1146	0.1071	**0.1155**	0.8%
	NDCG@5	0.0241	0.0261	0.0216	0.0115	0.0281	0.0326	0.0311	0.0348	**0.0386**	10.9%
	NDCG@20	0.0387	0.0403	0.0391	0.0198	0.0441	0.0483	0.0490	0.0505	**0.0551**	9.1%
Toys	HR@5	0.0463	0.0502	0.0274	0.0143	0.0526	0.0598	0.0542	0.0586	**0.0683**	14.2%
	HR@20	0.0941	0.0975	0.0688	0.0235	0.1038	0.1138	0.1132	0.1148	**0.1242**	8.2%
	NDCG@5	0.0306	0.0337	0.0174	0.0123	0.0362	0.0404	0.0297	0.0407	**0.0467**	14.7%
	NDCG@20	0.0441	0.0471	0.0291	0.0162	0.0506	0.0557	0.0464	0.0565	**0.0625**	10.6%

Table 2: Performance comparison of different methods. The best score in each row is in bold, and the second score is underlined. The last two columns are relative improvements compared to the best baseline results.



(a) Sports (b) Beauty (c) Toys

Figure 4: Performance comparison on ICMA different embedding dimension.

- **ICLRec** (Chen et al., 2022b): The model learns users' potential intentions from sequences through clustering and integrates the learned intentions into the model through assisted contrast loss.
- **IOCRec** (Li et al., 2023b): The model uses intent contrastive learning to solve the denoising problem in sequence recommendation tasks
- ***S^4*Rec** (Wei et al., 2024): The model uses online clustering to expertly group users according to their different potential intentions.

4.4 Performance Comparison

Table 2 shows the results of different methods on all datasets. We have the following observations. First, our model ICMA outperforms all baselines on all three datasets. The performance improvement ranges from 0.8% to 14.7%. In particular, ICLRec and IOCRec utilize the contrastive SSL task to learn the intention representation of SR, accurately modeling the user's preferences, which significantly improves performance. However, they are not as effective as ICMA, and one possible reason is that they use random data augmentation methods and do not use the substitute and insert

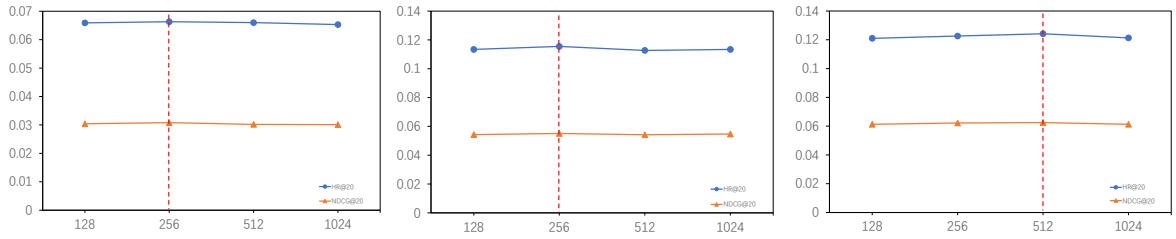
Model	Dataset					
	Sports		Beauty		Toys	
	HR	NDCG	HR	NDCG	HR	NDCG
(A) ICMA	**0.0663**	**0.0308**	**0.1155**	**0.0551**	**0.1242**	**0.0625**
(B) w/o S	0.0661	0.0306	0.1150	0.0546	0.1219	0.0624
(C) w/o I	0.0659	0.0305	0.1099	0.0529	0.1204	0.0614
(D) w/o Pos	0.0659	0.0303	0.1098	0.0541	0.1211	0.0599

Table 3: Ablation study of ICMA (HR@20 and NDCG@20).

augmentation operators that we introduce. These two operators can make better use of the relevance between items, enhance the diversity of data and the adaptability of the model. Among all the compared models, S^4Rec shows better performance than the other models, a possible reason is that online clustering is an important factor in improving the learning of user intent for sequential recommendation models. However, it is still less effective than our model, and a plausible explanation may be that it does not employ extrapolated position encoding, thus failing to adequately capture long-term dependencies in user behavior.

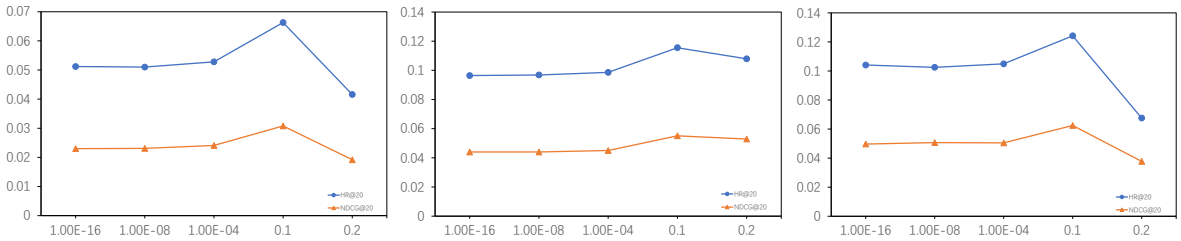
4.5 Ablation Study

We conduct ablation studies to verify the validity of the modules in our model and record our ex-



(a) Sports (b) Beauty (c) Toys

Figure 5: Comparison of ICMA performance with different cluster number K.



(a) Sports (b) Beauty (c) Toys

Figure 6: Performance comparison of ICMA w.r.t. different β.

perimental results in Table 3. From the table, we can see that by comparing (A) with (B) and (C), the performance of the model decreases in the absence of the " substitute" and "insert" operations, which illustrates the importance of utilizing item relevance. These operations introduce more item-relevant information and increase the diversity of the data, thus enabling the model to better capture user preferences. Comparing (A) and (D), We find that using extrapolated position encoding can effectively capture the long-term dependency of users and thus obtain further position information.

4.6 Hyper-parameter Analysis

We conduct experiments by tuning the embedding dimension, contrastive learning strength β, and the number of clusters K to identify the optimal hyperparameters for the ICMA model, using HR@20 and NDCG@20 as evaluation metrics.

As shown in Figure 4, the model performance improves with increasing embedding dimensions, reaching the best performance at 128 dimensions, after which it gradually declines. This suggests that excessively large embedding dimensions may introduce redundant information, increasing model complexity and leading to overfitting. Figure 5 shows that the optimal performance is achieved with a cluster number K of 256 on the Sports and Beauty datasets, with performance decreasing beyond 256. On the Toys dataset, the best performance is observed at $K = 512$, but further increasing K leads to a decline, likely due to excessive clustering making the data sparse and affecting the model's generalization ability. As seen in Figure 6, model performance significantly improves as β increases to 0.1, particularly between $\beta = 1e - 4$ and 0.1. When $\beta > 0.1$, performance declines, possibly because the positive and negative samples become too dispersed, affecting the model's generalization.

5 Conclusion

In this paper, we propose ICMA, a framework that improves traditional position encoding by extrapolating position encoding to more effectively capture long-term dependencies in user behavior. In addition, the model's adaptability and accuracy are enhanced by the design of a multi-view augmentation method, which fully utilizes item relevance and accurately reflects the user's intention. The experimental results on the three datasets show that ICMA has significant advantages over most baselines. Future work can further explore more augmentation operators based on item relevance to utilize more accurate and relevant information.

6 Limitations

Although ICMA shows performance improvements over most baseline methods, it is not without limitations. In some cases, item relevance may be weak,

or the relationship between user behavior and items may change frequently. In these situations, the operators may select inappropriate items, resulting in augmented data that does not match the actual context and introduces noisy data. Additionally, while ICMA effectively models long-term dependencies through extrapolated position encoding, it may overemphasize distant user behavior when the sequence length becomes too long, reducing attention to recent interactions.

Acknowledgments

This work was supported in part by the "20 New Universities" Project of Jinan City (202333023) and Shandong Provincial Natural Science Foundation (ZR2023QF006).

References

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

Renqin Cai, Jibang Wu, Aidan San, Chong Wang, and Hongning Wang. 2021. Category-aware collaborative sequential recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 388–397.

Lei Chen, Jingtao Ding, Min Yang, Chengming Li, Chonggang Song, and Lingling Yi. 2022a. Item-provider co-learning for sequential recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1817–1822.

Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022b. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*, pages 2172–2182.

Hanwen Du, Huanhuan Yuan, Pengpeng Zhao, Fuzhen Zhuang, Guanfeng Liu, Lei Zhao, Yanchi Liu, and Victor S Sheng. 2023. Ensemble modeling with contrastive knowledge distillation for sequential recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 58–67.

Yingpeng Du, Ziyan Wang, Zhu Sun, Yining Ma, Hongzhi Liu, and Jie Zhang. 2024. Disentangled multi-interest representation learning for sequential recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 677–688.

Ziwei Fan, Zhiwei Liu, Hao Peng, and Philip S Yu. 2023. Mutual wasserstein discrepancy minimization for sequential recommendation. In *Proceedings of the ACM Web Conference 2023*, pages 1375–1385.

Ziwei Fan, Zhiwei Liu, Jiawei Zhang, Yun Xiong, Lei Zheng, and Philip S Yu. 2021. Continuous-time sequential recommendation with temporal graph collaborative transformer. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 433–442.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. 2020. Supervised contrastive learning for pre-trained language model fine-tuning. *arXiv preprint arXiv:2011.01403*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

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

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE.

Dongjun Lee, Donggeun Ko, and Jaekwang Kim. 2023. Hierarchical contrastive learning with multiple augmentation for sequential recommendation. *arXiv preprint arXiv:2308.03400*.

Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. 2023a. Text is all you need: Learning language representations for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1258–1267.

Xuewei Li, Aitong Sun, Mankun Zhao, Jian Yu, Kun Zhu, Di Jin, Mei Yu, and Ruiguo Yu. 2023b. Multi-intention oriented contrastive learning for sequential recommendation. In *Proceedings of the sixteenth ACM international conference on web search and data mining*, pages 411–419.

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

Zhiwei Liu, Ziwei Fan, Yu Wang, and Philip S Yu. 2021b. Augmenting sequential recommendation with pseudo-prior items via reversely pre-training transformer. In *Proceedings of the 44th international ACM SIGIR conference on Research and development in information retrieval*, pages 1608–1612.

Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled self-supervision in sequential recommenders. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 483–491.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 43–52.

Xiuyuan Qin, Huanhuan Yuan, Pengpeng Zhao, Junhua Fang, Fuzhen Zhuang, Guanfeng Liu, Yanchi Liu, and Victor Sheng. 2023. Meta-optimized contrastive learning for sequential recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 89–98.

Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International conference on data mining*, pages 995–1000. IEEE.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Shuhan Wang, Bin Shen, Xu Min, Yong He, Xiaolu Zhang, Liang Zhang, Jun Zhou, and Linjian Mo. 2024. Aligned side information fusion method for sequential recommendation. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 112–120.

Shaowei Wei, Zhengwei Wu, Xin Li, Qintong Wu, Zhiqiang Zhang, Jun Zhou, Lihong Gu, and Jinjie Gu. 2024. Leave no one behind: Online self-supervised self-distillation for sequential recommendation. In *Proceedings of the ACM on Web Conference 2024*, pages 3767–3776.

Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 1259–1273. IEEE.

Yipeng Zhang, Xin Wang, Hong Chen, and Wenwu Zhu. 2023. Adaptive disentangled transformer for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3434–3445.

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 *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1893–1902.

Tianyu Zhu, Yansong Shi, Yuan Zhang, Yihong Wu, Fengran Mo, and Jian-Yun Nie. 2024. Collaboration and transition: Distilling item transitions into multi-query self-attention for sequential recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 1003–1011.