

Intent-Interest Disentanglement and Item-Aware Intent Contrastive Learning for Sequential Recommendation

Yijin Choi

UNIST

Ulsan, Republic of Korea

yjchoi0315@unist.ac.kr

Chiehyeon Lim*

UNIST

Ulsan, Republic of Korea

chlim@unist.ac.kr

Abstract

Recommender systems aim to provide personalized item recommendations by capturing user behaviors derived from their interaction history. Considering that user interactions naturally occur sequentially based on users' intents in mind, user behaviors can be interpreted as user intents. Therefore, intent-based sequential recommendations are actively studied recently to model user intents from historical interactions for a more precise user understanding beyond traditional studies that often overlook the underlying semantics behind user interactions. However, existing studies face three challenges: 1) the limited understanding of user behaviors by focusing solely on intents, 2) the lack of robustness in categorizing intents due to arbitrary fixed numbers of intent categories, and 3) the neglect of interacted items in modeling of user intents. To address these challenges, we propose Intent-Interest Disentanglement and Item-Aware Intent Contrastive Learning for Sequential Recommendation (IDCLRec). IDCLRec disentangles user behaviors into intents which are dynamic motivations and interests which are stable tastes of users for a comprehensive understanding of user behaviors. A causal cross-attention mechanism is used to identify consistent interests across interactions, while residual behaviors are modeled as intents by modeling their temporal dynamics through a similarity adjustment loss. In addition, without predefining the number of intent categories, an importance-weighted attention mechanism captures user-specific categorical intent considering the importance of intent for each interaction. Furthermore, we introduce item-aware contrastive learning which aligns intents that occurred the same interaction and aligns intent with item combinations occurred by the corresponding intent. Extensive experiments conducted on real-world datasets demonstrate the effectiveness of IDCLRec.

CCS Concepts

- Information systems → Recommender systems.

*Corresponding author

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

arXiv'25,

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-XXXX-X/2018/06

<https://doi.org/XXXXXX.XXXXXXX>

Keywords

Sequential Recommendation, Contrastive Learning, Intent-Interest Modeling

ACM Reference Format:

Yijin Choi and Chiehyeon Lim. 2025. Intent-Interest Disentanglement and Item-Aware Intent Contrastive Learning for Sequential Recommendation. In *Proceedings of (arXiv'25)*. ACM, New York, NY, USA, 14 pages. <https://doi.org/XXXXXX.XXXXXXX>

1 Introduction

Recommender systems are widely utilized in various fields such as e-commerce services to enhance user experiences by providing personalized item recommendations [3, 29]. The primary goal of recommender systems is to accurately predict users' preferences by leveraging their past interactions based on the assumption that their preferences can be inferred from historical data. Given that user interactions occur inherently sequential, many sequential recommendation models are developed to capture the temporal patterns of user behaviors. These models are designed to predict the next item that the user will interact with and have demonstrated good performance in recommendation tasks [5, 10, 20, 40, 48].

Traditional sequential recommendation models focus primarily on capturing behavioral patterns of users without delving into the underlying semantics behind the behaviors. Recently, considering that users make interactions under their intents in mind, intent-based sequential recommendations are actively studied with emphasizing the understanding of user behaviors from the perspective of intrinsic intentions [7, 17, 24, 35, 41]. In these approaches, user intents generally refer to the diverse and dynamic motivations that drive users to interact with items. These intents can be changed across various intent categories over time depending on the user's evolving purposes at each time step and the contextual attributes associated with items. Accordingly, these models aim to model intent representations that effectively capture the dynamic nature of user intents as behavior representations.

Such studies that interpret user behaviors as intents have shown remarkable performance. However, there are three challenges to be addressed to achieve a more comprehensive understanding of user behaviors and intents. First, focusing only on intentions may result in insufficient understanding of user behaviors and preferences. When users interact with items, they do not simply choose items that align solely with their immediate purposes but also take other factors into account. For instance, consider a user who recently purchased items under the intention of preparing clothes for travel as illustrated in Figure 1. Many travel-related items such as floral T-shirts or bright pants can be recommended based on the general purpose of traveling. However, a user can naturally have her own

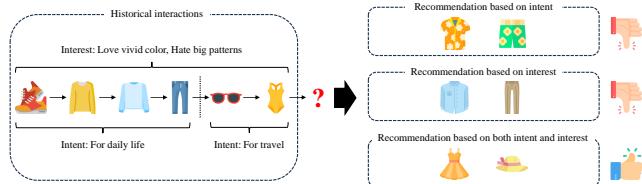


Figure 1: An example of user interactions and corresponding recommendation reflecting different semantic aspects in user behaviors.

personal tastes for items such that she loves vivid colors while hates big patterns. In this case, recommending floral T-shirts solely based on the travel intent may not be preferable by the user. Consequently, relying exclusively on the intent perspective may fail to fully capture the complexity of user behaviors. Second, explicitly fixing the number of intent categories is not a robust approach for capturing the categorical characteristics of intents. Intents are generally understood as categorical characteristics of user behaviors that apply across all users. Many studies cluster intent representations across all users or calculate the probability of each user’s intents belonging to specific intent categories to divide the overall intents into a predefined number of intent categories. Each intent representation is then aligned with the corresponding cluster or assigned to the category with the highest probability. However, the number of intent categories varies widely across studies, ranging from 3 to 1024 [7, 24, 26, 35, 47]. This significant variance suggests that arbitrarily dividing intents into a fixed number of categories lacks the robustness to accurately represent categorical intents. Lastly, overlooking interacted items may limit the understanding of user intents. Recent studies leverage contrastive learning to enhance intent representations by aligning augmented intent representations or bringing them closer to their assigned intent categories [7, 27, 35]. However, since these methods inevitably face the second challenge, these contrastive learning approaches may be ineffective by guiding intents to inaccurate categories. In this situation, even though interacted items that are the only explicit signals of user behaviors could play a crucial role in learning user intents, they are neglected in current studies.

To address these challenges, we propose Intent-Interest Disentanglement and Item-Aware Intent Contrastive Learning for Sequential Recommendation (IDCLRec) which introduces intent-interest disentanglement to tackle the first challenge and item-aware intent contrastive learning to tackle the second and third challenges. When we consider factors other than intents in user behaviors, it becomes evident that user interactions occur through both consistent behaviors related to users’ personal tastes across overall interactions in addition to time-varying behaviors associated with specific purposes which is intent [2, 9]. Although existing studies overlook these consistent properties of user behaviors, we capture them and call as interests. Then, by viewing user behaviors as the combination of intents and interests, we can explicitly disentangle user behaviors into intents and interests that are generally can be seen as long-term and short-term behaviors. Such disentanglement can lead to better understand of more precise behaviors and better learning of intent representations in further contrastive learning.

Furthermore, considering the personalized nature of recommendation tasks, we focus on capturing the categorical characteristics of individual user-level intents without predefining the number of intent categories. Given the categorical intent indicating a specific intent category, we incorporate combinations of items interacted with under the same intent category for each user in contrastive learning to provide valuable guidance for capturing nuanced intent representations. The detailed model structure of IDCLRec including these components is compared with existing studies in Table 1. Specifically, we design a causal cross-attention mechanism that identifies interests from user behaviors by reflecting the consistent characteristics of interests over time. The remaining behaviors after excluding interests are modeled as intents by incorporating the dynamic nature of intents through a similarity adjustment loss function that enforces similarity across adjacent time steps. Also, an importance-weighted attention mechanism is devised to learn categorical intents by integrating intent representations at different time steps that share high similarity with assigning higher importance to more similar intents. To further enhance the derived intent representation, we employ item-aware contrastive learning by considering user intents from both single-item and multi-item combination perspectives. The contrastive objectives align intent representations corresponding to the same target item and ensure that intent representation and item combinations interacted under the same intent category to be closer. Extensive experiments conducted on three real-world datasets demonstrate that IDCLRec significantly outperforms existing models, leading to more personalized and accurate recommendations through intent-interest disentanglement and user specific intent learning.

2 Related Works

2.1 Sequential Recommendation

Sequential recommendation aims to predict the next item that user will interact with by capturing the temporal patterns from users’ historical interactions [5, 20, 45]. Initial works on sequential recommendation often utilized markov chains that model user transitions between items as probabilistic processes [14]. With the development of deep learning, sequential recommendation models introduced neural networks to extract complex sequential patterns such as recurrent neural networks and convolutional neural networks-based models [16, 40]. In recent years, transformer-based models gained prominence with their powerful performance through attention mechanisms [10, 20, 37]. However, these models commonly focus on capturing behavioral patterns of users without considering the underlying semantics behind the behaviors.

2.2 Intent Recommendation

Recently, many studies focus on modeling user intents to understand user behaviors from an intent perspective for more effective recommendations [7, 17, 19, 21]. These studies can broadly be categorized into two groups which model implicit intent through utilizing side information and which model latent intent only from user interactions. Some studies introduce side information such as action types or item categories by considering that such side information provide valuable signals about user intents of interactions [4, 25, 41]. However, since side information is not always

Table 1: Component comparisons of the proposed model with existing models.

Model	User Behavior Semantic Perspective			Intent Separation	Contrastive Learning Perspective	
	General	Intent	Interest		Self-Representation	Cross-Representation
GRU4Rec (2015)	✓	✗	✗	–	–	–
SASRec (2018)	✓	✗	✗	–	–	–
BERT4Rec (2019)	✓	✗	✗	–	–	–
S ³ -RecMIP (2020)	✓	✗	✗	–	–	–
CL4SRec (2022)	✓	✗	✗	–	✓	✗
CoSeRec (2021)	✓	✗	✗	–	✓	✗
DuoRec (2022)	✓	✗	✗	–	✓	✗
DSSRec (2020)	✗	✓	✗	✓	–	–
ICLRec (2022)	✗	✓	✗	✓	✓	✗
IOCRec (2023)	✗	✓	✗	✓	✓	✗
ICSRec (2024)	✗	✓	✗	✓	✓	✗
IDCLRec	✗	✓	✓	✗	✓	✓

available, other studies try to model intents by reflecting only interactions [7, 35, 38]. These studies typically assign each intent to a predefined number of intent categories and employ contrastive learning in terms of intents to learn latent intents. Although they can capture intents well, their general approaches are insufficient in fully exploring the diverse characteristics of intents. Moreover, only focusing on intents also can lead to an underexploration of nuanced user behaviors.

2.3 Contrastive Learning

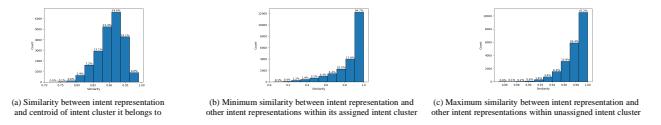
Contrastive learning which is a prominent self-supervised learning approach has achieved success in various domains such as computer vision [6, 13], natural language processing [11, 12], and recommender systems [48, 51]. It enhances representations learned from unlabeled data by giving signals by maximizing agreement between positive pairs which are typically derived as augmentations of the same data. The sequential recommendation studies with contrastive learning usually seek for effective data augmentation strategies or suitable positive representations to align with [28, 34]. While these approaches show good performance with enhancing representations, they often overlook the signals from items when contrasting representations.

3 Intent Analyses

In this paper where user behaviors are represented through both user interests and intents, we conduct intent analyses to capture the characteristics of intents that underlie interactions with items for comprehensive user intent modeling. To achieve this, we analyze intent representations obtained by ICSRec which is a current state-of-the-art (SOTA) model in intent recommendation to check its characteristics. Specifically, ICSRec learns a single unified intent representation for each user from their entire interaction sequence, implying that the entire sequence of interactions reflects a particular user intent. To categorize these intent representations, ICSRec assumes that there are K distinct intent categories applicable to users. Accordingly, it applies K-means clustering to group all users’ intent representations into K clusters. Each cluster ideally corresponds

to a particular type of intent containing similar intent representations and the cluster centroid serves as the representative vector for that intent category. Although this clustering does not explicitly provide a semantic label for each category, it can assign each user’s intent representation to a specific category characterized by a vector. Through learned intent representations, we examine the categorical and item-related characteristics of intents to get insights that inform the design of a model capable of more effectively capturing intrinsic intent features.

As noted in previous works, user intents can be understood as a categorical trait of behaviors. To dive into categorical nature of intents, we analyze the intents derived by ICSRec for Beauty dataset in terms of similarity distributions between intents as shown in Figure 2. Here, similarity is calculated as a sigmoid of the dot product considering the goal of recommendation task that suggests items with high similarity to users determined by the dot product. Specifically, Figure 2-(a) shows the distribution of similarity between each intent representation and centroid of cluster it belongs to. Where there are intents with high or relatively low similarity to the centroid of assigned clusters, such variation in similarity indicates that the degree to which each intent contributes to defining the intent category differs. In this context, the intents with higher similarity can be interpreted as more significant representation that define the intent category represented by the assigned cluster. Likewise, the distribution of the minimum similarity between each intent and other intent representations within the same cluster is presented in Figure 2-(b). The presence of low similarity with other intents for some intents means the existence of dissimilar intents within the same cluster. Such intents might be less important in representing the intent category compared to those with consistently high

**Figure 2: Similarity distributions between intent representations.**

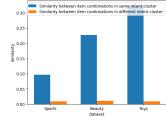


Figure 3: Average similarity between item combinations in same and different intent clusters.

similarity to other intents. Figure 2-(c) illustrates the distribution of the maximum average similarity for each intent by computing the average similarity between the intent and all other intents from each different cluster and identifying the highest average similarity among all clusters. This reveals that most intents exhibit high similarity with a specific cluster they do not belong to. Such high similarity suggests that intents share characteristics similar not only to their own intent category but also to other intent categories. Consequently, it can be inferred that the commonly used K-means clustering might not be effective in learning intent representations that distinctly capture specific intent categories. In summary, general clustering might not clearly group intents indicating a specific intent category and the reflection of relative importance of intents might be effective to capture intent categories.

We further investigate the relationship between intents and item combinations by analyzing the item sequences associated with each intent category. Figure 3 shows the average similarity of item sequences that output intent representations within the same cluster and across different clusters for Sports, Beauty, and Toys datasets. Each item sequence containing the items interacted with by each user is considered to indicate item combination that reflects a specific intent category by deriving a single intent representation. These sequences are represented as simple one-hot encoded vectors which assign 1 to interacted items and 0 to non-interacted items. The similarity between sequences is computed as the number of shared items using the dot product of these vectors. Although the similarity values are generally low due to the highly sparse nature of item sequences from the large number of items, the similarity of item sequences within the same cluster is much higher than the similarity of sequences across different clusters in all datasets. This higher similarity indicates that item sequences associated with similar intents tend to contain more overlapping items. In other words, a relationship exists between item combinations and intents such that item combinations interacted under the same intent category exhibit more similar patterns.

Based on these findings, the following conclusions can be drawn: 1) For the purpose of predicting individual preferences in recommendation tasks, it could be effective to extract similar intents for each user and learn categorical intents by reflecting their relative importance rather than separating all users' intents. 2) It could be effective to enhance intents in terms of items by considering the item combinations interacted with under users' current intent category for better alignment of item-level interactions and underlying user intents. Building upon these conclusions, we devise a model that explicitly incorporates modules for learning such beneficial intent characteristics, ensuring that the identified characteristics are effectively reflected in the learned representations.

4 Methodology

4.1 Problem Definition

Let \mathcal{U} and \mathcal{V} denote set of users and items, respectively. For each user $u \in \mathcal{U}$, the user interaction history is represented as chronologically ordered sequence of items $S^u = [v_1^u, v_2^u, \dots, v_{|S^u|}^u]$ where $v_t^u \in \mathcal{V}$ is the item that user u interacted with at the t -th time step and $|S^u|$ is the number of interacted items for user u . To standardize length of sequences, all user sequences are truncated or padded to satisfy a fixed maximum length N . Specifically, if $|S^u| \geq N$, the most recent N interacted items are considered as $S^u = [v_{|S^u|-N+1}^u, \dots, v_{|S^u|}^u]$. Otherwise, if $|S^u| < N$, we pad the sequence with a special padding item $v_0 \notin \mathcal{V}$ at the beginning until the length of sequence reaches N as $S^u = [v_0, \dots, v_0, v_1^u, \dots, v_{|S^u|}^u]$. The goal of sequential recommendation is to predict the next item $v_{|S^u|+1}^u$ that user u is most likely to interact with at the time step $|S^u| + 1$ given historical interaction sequence S^u , which can be formulated as follows:

$$\operatorname{argmax}_{v_i \in \mathcal{V}} P(v_{|S^u|+1}^u = v_i | S^u). \quad (1)$$

4.2 Overview

The overall framework of IDCLRec is shown in Figure 4. The model aims to fully understand user behaviors by exploring both user interest and intent perspectives. The architecture consists of three principal components: interest representation learning, intent representation learning, and contrastive learning for intent representation. First, the sequence encoder learns user behavior representations from interaction history. To capture interests, we devise a causal cross-attention mechanism that reflects the consistent characteristics of interests, allowing the model to extract interests from sequence representations. We then introduce a loss function to learn intents by focusing on dynamic characteristics. We also perform an importance-weighted attention which leverages the similarity between intents for intent extraction and aggregation to express categorical characteristics of user intents. Finally, two contrastive learning tasks are introduced to further enhance the intent representation in terms of items. The first contrastive task aligns intent representations of sequences sharing the same target item and the second brings the intent representation closer to the items interacted with under the corresponding intent. By combining the learned interest and intent representations, the model predicts the next item a user is likely to interact with.

4.3 Sequence Representation Learning

Transformer has been extensively used in sequential recommendation tasks to learn user sequence representations. Especially, SASRec [20] stands out as a representative model due to its effective utilization of the transformer encoder architecture. Recent studies have frequently employed SASRec for representation extraction by leveraging its powerful capabilities [35, 44, 49, 51]. Motivated by these strengths, we adopt SASRec as an encoder for sequence representation learning.

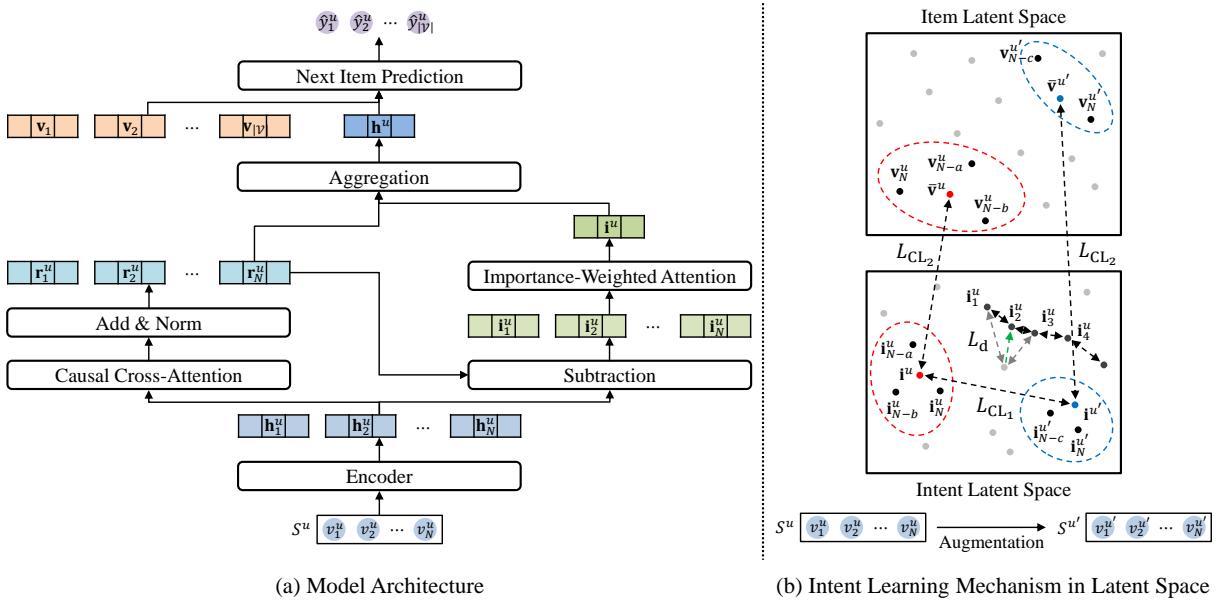


Figure 4: Overview of the proposed IDCLRec. (a) illustrates the architecture of IDCLRec including the modules for each process. (b) presents the mechanism of the training loss function specifically designed for intent representation learning.

4.3.1 Embedding Layer. For the item set \mathcal{V} , we construct an item embedding matrix $\mathbf{V} \in \mathbb{R}^{(|\mathcal{V}|+1) \times d}$ where d represents the dimensionality of the embedding vectors. The additional dimension in \mathbf{V} is allocated for a padding item v_0 . To incorporate the positional information of items, we build a position embedding matrix $\mathbf{P} \in \mathbb{R}^{N \times d}$. Then, given a user sequence S^u , sequence embedding $E^u \in \mathbb{R}^{N \times d}$ is computed by adding the item embedding \mathbf{V}^u with the corresponding position embedding \mathbf{P} as follows:

$$E^u = \text{Embedding}(S^u) = \mathbf{V}^u + \mathbf{P} = [v_1^u + p_1, v_2^u + p_2, \dots, v_N^u + p_N], \quad (2)$$

where $v_t^u \in \mathbb{R}^d$ is the embedding of item v_t^u which is interacted with at the t -th time step in the sequence and $p_t \in \mathbb{R}^d$ is the positional embedding corresponding to the t -th time step.

4.3.2 Sequence Representation Learning Layer. After obtaining the sequence embedding, we employ a sequence encoder to learn sequence representation that captures the evolving behavior patterns of user over time. Formally, we define the sequence encoder $f(\cdot)$ which transforms the sequence embedding into a sequence representation $H^u \in \mathbb{R}^{N \times d}$ as follows:

$$H^u = f(E^u), \quad (3)$$

where $f(\cdot)$ denotes the SASRec model in this study. The $H^u = [h_1^u, h_2^u, \dots, h_N^u]$ indicates user behaviors across all time steps where each vector $h_t^u \in \mathbb{R}^d$ represents the user's behavioral representation at time step t , reflecting interaction history until time step t .

4.4 Interest Representation Learning

User behaviors can be composed of two main components: inherent interests and dynamic intentions. Interests represent unique characteristics of users which tends to remain consistent across different

time steps. On the other hand, intentions capture the contextual motivations that can be changed at each time step. To disentangle these components from the user behaviors, we focus on their distinct properties. Specifically, recognizing that interests emerge as stable user behavior patterns across the user interaction history, we introduce a causal cross-attention mechanism to extract interests from the user behaviors.

4.4.1 Causal Cross-Attention Mechanism. The causal cross-attention mechanism aims to learn interest representation by focusing on temporal dependencies in user behaviors. By extracting similar information between the user's current and previous interaction behaviors, the model captures consistent patterns across user behaviors.

First, we construct a padded version of the original sequence representation H^u denoted as $H_{\text{pad}}^u = [\bar{0}, h_1^u, h_2^u, \dots, h_N^u]$ where $\bar{0} \in \mathbb{R}^d$ is a zero vector for padding. The user behaviors at the current and previous time steps are represented as $H_{\text{cur}}^u = [h_1^u, h_2^u, \dots, h_N^u]$ and $H_{\text{pre}}^u = [\bar{0}, h_1^u, \dots, h_{N-1}^u]$, respectively. To capture similarities between these behaviors, we employ a causal cross-attention mechanism based on scaled dot-product attention [43] which is formulated as:

$$\hat{\mathbf{R}}^u = \text{Attention}(\mathbf{Q}_1, \mathbf{K}_1, \mathbf{V}_1) = \text{softmax} \left(\frac{\mathbf{Q}_1 \mathbf{K}_1^T}{\sqrt{d}} + \mathbf{M} \right) \mathbf{V}_1. \quad (4)$$

$\mathbf{Q}_1 = H_{\text{cur}}^u \mathbf{W}^{\mathbf{Q}_1} + \mathbf{b}^{\mathbf{Q}_1}$, $\mathbf{K}_1 = H_{\text{pre}}^u \mathbf{W}^{\mathbf{K}_1} + \mathbf{b}^{\mathbf{K}_1}$, and $\mathbf{V}_1 = H_{\text{pre}}^u \mathbf{W}^{\mathbf{V}_1} + \mathbf{b}^{\mathbf{V}_1}$ are the projected behavior representations through linear transformations where $\mathbf{W}^{\mathbf{Q}_1}, \mathbf{W}^{\mathbf{K}_1}, \mathbf{W}^{\mathbf{V}_1} \in \mathbb{R}^{d \times d}$ are weight matrices and $\mathbf{b}^{\mathbf{Q}_1}, \mathbf{b}^{\mathbf{K}_1}, \mathbf{b}^{\mathbf{V}_1} \in \mathbb{R}^d$ are bias terms which are introduced to handle the padding vector. To preserve the nature of sequential behaviors, we apply a causal masking by adding the mask matrix $\mathbf{M} \in \mathbb{R}^{N \times N}$

where each element M_{ij} is defined as follows:

$$M_{ij} = \begin{cases} 0 & \text{if } i \geq j \\ -\infty & \text{if } i < j \end{cases}. \quad (5)$$

The causal masking ensures that the attention mechanism focuses only on the behaviors at previous time steps and prevents information leakage from future behaviors. Then, the term $\text{softmax}\left(\frac{\mathbf{Q}_1 \mathbf{K}_1^T}{\sqrt{d}} + \mathbf{M}\right)$ indicates the attention weights that quantify the relevance between the current and each of its previous behaviors. Applying these attention weights to \mathbf{V}_1 allows the model to selectively integrate information from previous behaviors that are most relevant to the current behavior. Consequently, the resulting representation $\hat{\mathbf{R}}^u \in \mathbb{R}^{N \times d}$ contains consistent information over user behaviors where $\hat{\mathbf{r}}_t^u \in \mathbb{R}^d$ denotes information about user behaviors from time step 1 to $t-1$ with considering their relevance with the behavior at step t .

In this step, we apply cross-attention rather than self-attention to reduce the bias that can arise from self-attention as behavior at each time step attends not only to previous steps but also to itself. This bias can overemphasize current behavior and overshadow valuable information from previous behaviors. In contrast, the introduced cross-attention focuses solely on previous time steps, allowing the model to avoid this bias and better capture consistent patterns in user behavior. This approach supports our goal by emphasizing relationships between current and previous behaviors.

4.4.2 Residual Connection and Layer Normalization. While the causal cross-attention mechanism effectively reveals consistent patterns by aggregating previous behaviors similar to the current behavior, it excludes the current behavior itself. To address this, we apply a residual connection that reintroduces the current behavior representation \mathbf{h}_t^u into the attention output $\hat{\mathbf{r}}_t^u$ for each time step t . By incorporating the residual connection, the model retains crucial information about the current behavior with maintaining the information learned from previous behaviors.

Our goal in this section is to learn a user interest representation by focusing on similar patterns in user behaviors. Since \mathbf{h}_t^u contains both similar and different information compared to $\hat{\mathbf{r}}_t^u$, simply adding them can disrupt the learned patterns. Hence, we add a layer normalization [1] to implicitly emphasize only the similar features of \mathbf{h}_t^u . Layer normalization normalizes the representation along the feature dimension, preventing the unique patterns of \mathbf{h}_t^u from being emphasized and enhancing similar patterns between \mathbf{h}_t^u and $\hat{\mathbf{r}}_t^u$. It allows the final representation to capture information of \mathbf{h}_t^u aligned with learned consistent information $\hat{\mathbf{r}}_t^u$. Therefore, the final representation can exhibit the consistent user behavior patterns from time step 1 to t , thereby representing the user's interest.

Overall, the interest representation $\mathbf{R}^u \in \mathbb{R}^{N \times d}$ is derived as follows:

$$\mathbf{R}^u = \text{LayerNorm}\left(\hat{\mathbf{R}}^u + \mathbf{H}^u\right), \quad (6)$$

where $\mathbf{r}_t^u \in \mathbb{R}^d$ is the element of \mathbf{R}^u which represents the user's interest representation at time step t .

4.5 Intent Representation Learning

Based on the view that user behaviors are composed of interests and intentions, we derive the intent representation as the residual obtained by subtracting the interest representation from the behavior representation, denoted as:

$$\mathbf{I}^u = \mathbf{H}^u - \mathbf{R}^u. \quad (7)$$

$\mathbf{I}^u = [\mathbf{i}_1^u, \mathbf{i}_2^u, \dots, \mathbf{i}_N^u] \in \mathbb{R}^{N \times d}$ represents user intents across all time steps where $\mathbf{i}_t^u \in \mathbb{R}^d$ denotes the user's intent representation at time step t . To properly represent user intents, we design an auxiliary loss and model structure that account for both the evolving intent over time and its categorical characteristics.

4.5.1 Dynamic Intent Representation with Auxiliary Loss. Intents fluctuate at each time step due to the influence of changing interaction purposes and attributes of each item the user interacts with. However, since multiple interactions can occur with a consistent purpose in a short period, intents can exhibit similar patterns across adjacent time steps. Thus, one key property of intents is continuous changing over time while maintaining similarities between intents at consecutive time steps. In a situation where the model structurally captures the characteristics of interests, intents which are derived by subtracting interests from the behavior representation may indirectly retain some of its characteristics. However, intents are not explicitly modeled within the current architecture.

To directly reflect the dynamic nature of intents, we design a loss function L_d that minimizes the difference between consecutive intent representations by increasing their similarity:

$$L_d = \frac{1}{N-1} \sum_{t=1}^{N-1} (1 - \text{sigmoid}(\text{sim}(\mathbf{i}_t^u, \mathbf{i}_{t+1}^u))), \quad (8)$$

where $\text{sim}(\cdot)$ represents the similarity measure which is determined as the dot product in this study. This loss function allows intents to change gradually across time steps while maintaining coherence between consecutive intents. By introducing this auxiliary loss function, the model is guided to learn not only user interests but also the underlying characteristics of user intents, balancing stability and change over time.

4.5.2 Categorical Intent Representation with Importance-Weighted Attention. Users interact with items under various intents which can be viewed as a categorical characteristic of their behaviors. These intent categories represent distinct motivations that influence interactions, meaning that users interact with items based on specific intents belonging to the intent categories. Consequently, intents that are similar across different time steps can be grouped into the same intent category, reflecting a common underlying purpose. In such a situation, rather than focusing on each individual intent representation, the representation of the assigned intent category can reveal a more precise user intent for interactions by indicating a coherent intent category that encompasses all similar intents. Thus, to learn intent category that will determine the users' next interactions, we assume that users will interact with items under the specific intent category that their most recent intent is assigned. Under this assumption, we enhance the representation of the most recent intent by aggregating similar intents and derive the representation of intent category. This categorization of intents can allow

to generalize individual intents and better understand the users' underlying motivations. However, since the detailed relevance of these intents can vary, simple aggregation may not be sufficient. Some intents may exhibit strong unique characteristics specific to their own intent category while others may also be closely related to intents from different categories. To address these differences in relevance, we devise a intent importance extraction module which employs an attention mechanism over the most recent intent and its similar intents to focus on information from the more important intents.

We begin by identifying the intent representations from previous time steps that share similar characteristics with the most recent intent. To achieve this, we compute the similarity between the most recent intent representation i_N^u and each of the previous intent representations, i_t^u to i_{N-1}^u . Based on the computed similarities, we filter out the intent representations whose similarity to i_N^u exceed a predefined threshold. The set of similar intent representations, denoted as i_{sim}^u , is formally defined as:

$$i_{\text{sim}}^u = \{i_t^u \mid \text{sim}(i_N^u, i_t^u) \geq \delta, 1 \leq t \leq N-1\} \cup \{i_N^u\}, \quad (9)$$

where δ is the threshold that controls the degree of similarity required for intents to be considered similar. The representation matrix for the selected similar intent representations is then defined as $\tilde{\mathbf{I}}^u = [i_t^u \mid i_t^u \in i_{\text{sim}}^u] \in \mathbb{R}^{|i_{\text{sim}}^u| \times d}$.

To learn an intent representation that can accurately reflect its associated intent category, the model should focus on important intents that have strong relevance to other similar intents. Accordingly, motivated by importance extraction module introduced by [33], we compute the relevance scores between intents, denoted as $\mathbf{S} \in \mathbb{R}^{|i_{\text{sim}}^u| \times |i_{\text{sim}}^u|}$, using a modified self-attention mechanism as follows:

$$\mathbf{S} = \text{sigmoid} \left(\frac{\mathbf{Q}_2 \mathbf{K}_2^T}{\sqrt{d}} \right), \quad (10)$$

where $\mathbf{Q}_2 = \tilde{\mathbf{I}}^u \mathbf{W}^{\mathbf{Q}_2}$ and $\mathbf{K}_2 = \tilde{\mathbf{I}}^u \mathbf{W}^{\mathbf{K}_2}$ are obtained by passing the intent representations through linear transformations. Here, $\mathbf{W}^{\mathbf{Q}_2} \in \mathbb{R}^{d \times d}$ and $\mathbf{W}^{\mathbf{K}_2} \in \mathbb{R}^{d \times d}$ are weight matrices. Instead of using the softmax function which normalizes the relevance score across different intents, we use the sigmoid function to provide an objective measure of the relevance between each pair of intents.

As intent representations that are more relevant to other intents within the same category are important for expressing the intent category, the importance of each intent can be measured by its relevance scores with other intents. To compute the intent importance, we first sum the relevance scores between each intent and other intents whereas excluding itself to avoid high scores between identical intents from distorting the importance computation. We then apply the softmax function to these summed scores to compute the final importance weights for each intent. The importance weight for the i -th intent in i_{sim}^u is computed as follows:

$$\beta_i = \frac{\exp \left(\sum_{j=1, j \neq i}^{|i_{\text{sim}}^u|} S_{ij} \right)}{\sum_{k=1}^{|i_{\text{sim}}^u|} \exp \left(\sum_{j=1, j \neq k}^{|i_{\text{sim}}^u|} S_{kj} \right)}, \quad (11)$$

where $1 \leq i \leq |i_{\text{sim}}^u|$ and S_{ij} represents the relevance score between the i -th and j -th intent representations in i_{sim}^u .

The user's final intent representation $i^u \in \mathbb{R}^d$ is computed as a weighted sum of the selected intent representations where the weights reflect the importance of each intent:

$$i^u = \sum_{i=1}^{|i_{\text{sim}}^u|} \beta_i \tilde{i}_i^u, \quad (12)$$

where $\tilde{i}_i^u \in \mathbb{R}^d$ is the i -th intent representation in i_{sim}^u . This importance-weighted categorical intent representation captures the comprehensive intent of the user at the most recent time step, N , by selectively focus on the important intents that strongly indicate the user's categorical intent among the most recent intent and its similar intents from previous time steps. From the aforementioned assumption about user intents at the next time step, i^u which indicates the intent category of the most recently interacted intent can serve as the intent representation to guide user behaviors for the next interaction.

4.6 Intent Contrastive Learning

To further capture the nuanced intent representation, we leverage contrastive learning by focusing on the items which users interact. The interactions which occurred by users' preferences can provide critical insights about their underlying intents. Hence, considering the items that users interact with is essential for understanding their intents. To identify the targets for contrastive learning, we apply sequence augmentation in terms of items by randomly selecting a sequence that shares the same target item as the original sequence. For both the original sequence and the augmented sequence, we derive categorical intent representations. By employing contrastive learning, we aim to refine the intent representation from two perspectives related to the items that users interact with. Basically, the contrastive loss function based on InfoNCE [32] is used to minimize the distance to positive sample to be closer while maximizing the distance to negative samples to be far apart. This loss function for a given anchor \mathbf{x} is defined as follows:

$$L_{\text{CL}}(\mathbf{x}, \mathbf{x}_{\text{pos}}) = -\log \frac{\exp(\text{sim}(\mathbf{x}, \mathbf{x}_{\text{pos}})/\tau)}{\exp(\text{sim}(\mathbf{x}, \mathbf{x}_{\text{pos}})/\tau) + \sum_{\mathbf{x}_n \in \mathcal{X}_{\text{neg}}} \exp(\text{sim}(\mathbf{x}, \mathbf{x}_n)/\tau)}, \quad (13)$$

where $(\mathbf{x}, \mathbf{x}_{\text{pos}})$ represents a pair of positive sample representations and $(\mathbf{x}, \mathbf{x}_n)$ represents a pair of negative sample representations. \mathbf{x}_n is drawn from the set of negative samples \mathcal{X}_{neg} and τ is the temperature parameter.

4.6.1 Intent-Intent Contrastive Learning. The augmented sequence $S^{u'}$ is the interaction history of user u' with the target item as v_{N+1}^u . The intent representation of this sequence $i^{u'}$ is derived through the described model architecture. In this context, intent representations for the original and augmented sequences can be treated as positive pairs based on the rationale that users who interact with the same item may share the same intent. Therefore, intent representations for sequences with the same target item should be closer to each other in the latent space.

To achieve this, we apply contrastive learning that brings the intent representations i^u and $i^{u'}$ closer, while intent representations from sequences with different target items far apart. The contrastive

loss function L_{CL_1} is defined as:

$$L_{CL_1} = L_{CL}(\mathbf{i}^u, \mathbf{i}^{u'}) + L_{CL}(\mathbf{i}^{u'}, \mathbf{i}^u), \quad (14)$$

where $(\mathbf{i}^u, \mathbf{i}^{u'})$ is a positive pair and X_{neg} contains intent representations from sequences with different target items. In detail, for a batch size of $|B|$, there are $2|B|$ intent representations after sequence augmentation. Among the $2(|B| - 1)$ potential negative samples in X_{neg} , some representations might be derived from sequences that share the same target item as the positive pair. These are considered as false negative samples and are excluded from X_{neg} to ensure that only intent representations from sequences with different target items are included in the negative sample set.

Conversely, users interacting with the same item may have different intentions. Considering the observed relationship between items and intents, this intention can be inferred from the combination of multiple items interacted with under similar intents. While individual items can be associated with various intents, the combination of items can provide a stronger signal about the specific intent behind interactions. Therefore, we directly adjust user intents through contrastive learning with the aggregated representations of items interacted with under intents that constitute the final intent representation.

The intent representation \mathbf{i}^u is derived from the intent representations in i_{sim}^u . Considering these intent representations, we define the item representation matrix $\mathbf{V}_{sim}^u \in \mathbb{R}^{|i_{sim}^u| \times d}$ which contains the representations of items interacted with under the intents in i_{sim}^u , as follows:

$$\mathbf{V}_{sim}^u = [\mathbf{v}_t^u \mid t \in \{t \mid \mathbf{i}_t^u \in i_{sim}^u\}]. \quad (15)$$

To aggregate these item representations, we simply average them and treat the centroid of the item representations as a representation for the combination of interacted items. The centroid $\bar{\mathbf{v}}^u \in \mathbb{R}^d$ is calculated as:

$$\bar{\mathbf{v}}^u = \frac{1}{|\mathbf{V}_{sim}^u|} \sum_{\mathbf{v}_t^u \in \mathbf{V}_{sim}^u} \mathbf{v}_t^u. \quad (16)$$

This item centroid serves as the positive sample for intent representation for alignment between intent and the items that reflect it. We then apply contrastive learning between the intent representations \mathbf{i}^u and $\mathbf{i}^{u'}$ and the corresponding centroids of their respective item representations $\bar{\mathbf{v}}^u$ and $\bar{\mathbf{v}}^{u'}$. The contrastive loss function L_{CL_2} is designed to bring each intent representation closer to the centroid of the item representations that were interacted under the corresponding intent as follows:

$$L_{CL_2} = L_{CL}(\mathbf{i}^u, \bar{\mathbf{v}}^u) + L_{CL}(\bar{\mathbf{v}}^u, \mathbf{i}^u) + L_{CL}(\mathbf{i}^{u'}, \bar{\mathbf{v}}^{u'}) + L_{CL}(\bar{\mathbf{v}}^{u'}, \mathbf{i}^{u'}). \quad (17)$$

Here, contrastive learning is conducted separately for the original sequences and the augmented sequences with $(\mathbf{i}^u, \bar{\mathbf{v}}^u)$ and $(\mathbf{i}^{u'}, \bar{\mathbf{v}}^{u'})$ representing positive pairs for the original and augmented sequences, respectively. We omit false negative removal since the intent representation is learned to be specialized for each user based on their unique interaction history. Hence, all $2(|B| - 1)$ intent and centroid representations from other sequences are treated as negatives and contained in X_{neg} .

4.7 Multi-Task Learning

4.7.1 User Representation Learning. For obtained interest representation and intent representation, we aggregate two representations simply by addition to present a user representation. In this way, we assume that the relative contribution of user interest \mathbf{r}_N^u and user intent \mathbf{i}^u are just same on representing user behavior at time step N . Then, the final user representation $\mathbf{h}^u \in \mathbb{R}^d$ is computed as a combination of the user interest and intent representations as follows:

$$\mathbf{h}^u = \mathbf{r}_N^u + \mathbf{i}^u. \quad (18)$$

4.7.2 Next Item Prediction. After deriving the user representation, the model predicts how much a user would prefer each item in the candidate item set \mathcal{V} . To do this, we compute the dot product between the user representation and the item representation. The user preference score \hat{y}_i^u for item v_i is computed as follows:

$$\hat{y}_i^u = \mathbf{v}_i^T \mathbf{h}^u, \quad (19)$$

where \mathbf{v}_i is the item representation for item v_i drawn from \mathcal{V} . As before, the dot product works as a similarity measure between the user and the item to indicate how well the item matches the user's preferences. This allows the model to compute preference scores for all items in the candidate set.

For the general recommendation task, the goal of the model is to assign the highest preference score to the ground-truth item v_g^u which represents the item that the user interacted with at the next time step $N+1$. Therefore, we train the model using a cross-entropy loss function which encourages the preference score of the ground-truth item \hat{y}_g^u to be higher than the scores of all other items in \mathcal{V} . The training loss L_{rec} is formulated as follows:

$$L_{rec} = -\frac{\exp(\hat{y}_g^u)}{\sum_{v_i \in |\mathcal{V}|} \exp(\hat{y}_i^u)}. \quad (20)$$

4.7.3 Model Training. We train the model using a multi-task learning strategy to jointly optimize three tasks through their corresponding objective functions: the recommendation task via Eq. (20), the adjustment task via Eq. (8), and the contrastive learning tasks via Eq. (14) and Eq. (17). Formally, the total training loss L is defined as a weighted sum of these individual objectives as follows:

$$L = L_{rec} + \lambda_d L_d + \lambda_{CL_1} L_{CL_1} + \lambda_{CL_2} L_{CL_2}, \quad (21)$$

where λ_d , λ_{CL_1} , and λ_{CL_2} are hyperparameters that control the relative contributions of the adjustment and two contrastive learning tasks, respectively.

5 Experiments

5.1 Experimental Settings

5.1.1 Datasets. We evaluate the effectiveness of our proposed model on three public datasets collected from a real-world platform. Specifically, we utilize the Amazon Review Dataset [15, 31] which presents user review data collected from Amazon which is one of the largest e-commerce platform. This dataset can be divided into various subsets based on item categories. Following many existing studies in contrastive learning-based recommendation and intent-based recommendation [7, 34, 35, 50], in this paper, we adopt three representative subcategories: "Sports and Outdoors", "Beauty", and "Toys and Games".

For high quality of datasets, we preprocess the datasets following common practice in [7, 36, 48]. All interactions are considered as implicit feedback which treat all observed interactions as positive feedback while others as negative regardless of explicit numeric ratings. Then, we keep the “5-core” datasets which eliminate users and items with less than five interactions. For model training and evaluation, we divide the datasets into training, validation, and test datasets. Specifically, we chronologically sort the historical items for each user according to the interaction timestamps to construct the user’s interaction sequence. Following recent studies that contain data augmentation through sequence splitting in implementation and validate its effectiveness [26, 35, 36, 39, 52], we split each user sequence into subsequences and augment the data with these additional subsequences. For each user sequence, the last interacted item is used as the target for test where the remaining interactions are considered as the input sequence. Similarly, the second-to-last item is used for validation and the other items are used for training. Table 2 summarizes the statistical details of these datasets after preprocessing.

5.1.2 Evaluation Metrics. To ensure a fair model evaluation, we predict the user preference scores for items in the whole item set without negative sampling [23, 46]. Then, the performance for the model prediction is evaluated through top- k Hit Ratio (HR@ k) and Normalized Discounted Cumulative Gain (NDCG@ k) [18] which are commonly used evaluation metrics in top- k recommendation. In short, HR measures whether the ground-truth item presents in top- k item list derived by model and NDCG measures how well the predicted order of items aligns with the actual order of items in user sequence. We report HR and NDCG with $k \in 5, 10, 20$.

5.1.3 Baselines. To verify the effectiveness of our model, we compare our model with the following representative baselines belonging to three groups of sequential recommendations:

- **General sequential models:** GRU4Rec [16] applies gated recurrent unit to model user sequences. SASRec [20] employs an uni-directional self-attention mechanism to capture sequential patterns in user sequences, becoming one of the SOTA model in sequential recommendation task.
- **Sequential models with self-supervised learning:** BERT4Rec [37] adopts a bi-directional self-attention mechanism and performs a Cloze task which predicts masked items instead of the next item, inspired by BERT [8]. S³-Rec_{MIP} [50] introduces self-supervised learning to capture correlations between items, attributes, and sequences. Since all our model and other baselines do not consider item attribute information, we only contain masked item prediction module for fair comparison, namely S³-Rec_{MIP}. CL4SRec [48] proposes three data augmentation methods and first applies contrastive

Table 2: Statistics of the datasets.

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

learning between original and augmented sequences for sequential recommendation task. CoSeRec [28] further devises two data augmentation strategies based on item correlations for robustness of augmented sequence. DuoRec [36] introduces an unsupervised model-level augmentation method using dropout and a supervised positive sampling strategy that accounts for semantic similarity between sequences.

- **Sequential models considering intent:** DSSRec [30] proposes a sequence-to-sequence training strategy that contains self-supervision and intent disentanglement. ICLRec [7] performs clustering on intent representations and contrastive learning that reflects the learned intent clusters for latent intent modeling. IOCRec [26] designs intent-level contrastive learning to emphasize the main intent among multiple intents for user behavior. ICSRec [35] further improves ICLRec by leveraging data augmentation considering subsequences and sequence augmentation considering target items.

5.1.4 Implementation Details. We implement BERT4Rec, S³-Rec_{MIP}, CoSeRec, DuoRec, ICLRec, IOCRec, and ICSRec using official codes provided by authors. GRU4Rec, SASRec, CL4SRec, and DSSRec are implemented based on public resources. For all baselines, we follow the optimal hyperparameter combinations reported in their original papers. Our model is implemented in PyTorch. For IDCLRec, the sequence encoder consists of 2 number of self-attention blocks and attention heads. The embedding dimension d is set as 64 and the maximum sequence length N is 50. We apply a dropout rate of 0.5 and set the temperature parameter τ as 1. The model is optimized using Adam optimizer [22] with a learning rate of 0.001 and a batch size of 256. The model is trained for up to 300 epochs with applying early stopping which stops training if the NDCG@20 on the validation dataset does not improve for 40 consecutive epochs. For hyperparameter tuning, we search for the best combination of the intent similarity threshold δ within {0.5, 0.6, 0.7, 0.8, 0.9} and weights for each loss component λ_d , λ_{CL_1} , and λ_{CL_2} within {0.1, 0.2, 0.3, 0.4, 0.5} through grid search. All experiments are conducted using three different random seeds and the average results are reported to ensure robustness of models.

5.2 Performance Comparison

Table 3 shows the performance of the proposed model and all baselines across three datasets. In summary, our proposed IDCLRec significantly outperforms other baselines on all metrics and datasets.

Specifically, for general sequential models, SASRec shows better performance than GRU4Rec suggesting that the attention mechanism in SASRec can capture sequential patterns in user behaviors more effectively than RNN-based GRU4Rec. Although BERT4Rec adopts a self-supervised learning strategy in addition to the attention mechanism, it sometimes performs better or worse than SASRec. These inconsistent results might arise as the masked item prediction in BERT4Rec does not fit well with the sequential recommendation task that aims to predict the next item whereas the bi-directional attention mechanism well captures behavior dependencies. S³-Rec_{MIP} which also employs masked item prediction performs poorly in all situations. While S³-Rec includes several modules that utilize additional item attribute information, we do not consider these modules to exclude attribute information. In

Table 3: Performance comparisons of the proposed model with baselines where best results are boldfaced and second-best results are underlined.

Dataset	Metric	GRU4Rec	SASRec	BERT4Rec	S^3 -Rec _{MIP}	CL4SRec	CoSeRec	DuoRec	DSSRec	ICLRec	IOCRec	ICSRec	IDCLRec
Sports	HR@5	0.0101	0.0191	0.0226	0.0119	0.0246	0.0271	0.0326	0.0195	0.0274	0.0294	<u>0.0383</u>	0.0431
	HR@10	0.0176	0.0300	0.0354	0.0199	0.0389	0.0404	0.0502	0.0309	0.0425	0.0467	<u>0.0549</u>	0.0618
	HR@20	0.0293	0.0462	0.0536	0.0326	0.0596	0.0599	0.0744	0.0461	0.0640	0.0692	<u>0.0766</u>	0.0878
	NDCG@5	0.0063	0.0125	0.0151	0.0076	0.0162	0.0183	0.0192	0.0130	0.0179	0.0168	<u>0.0270</u>	0.0299
	NDCG@10	0.0087	0.0160	0.0192	0.0102	0.0207	0.0226	0.0249	0.0167	0.0227	0.0223	<u>0.0323</u>	0.0360
	NDCG@20	0.0117	0.0201	0.0237	0.0134	0.0260	0.0275	0.0310	0.0205	0.0281	0.0279	<u>0.0378</u>	0.0426
Beauty	HR@5	0.0169	0.0349	0.0390	0.0172	0.0396	0.0482	0.0590	0.0391	0.0504	0.0515	<u>0.0677</u>	0.0714
	HR@10	0.0296	0.0545	0.0602	0.0300	0.0636	0.0712	0.0875	0.0615	0.0727	0.0816	<u>0.0932</u>	0.1013
	HR@20	0.0487	0.0843	0.0900	0.0454	0.0950	0.1032	0.1243	0.0918	0.1055	0.1179	<u>0.1284</u>	0.1399
	NDCG@5	0.0105	0.0223	0.0251	0.0106	0.0255	0.0319	0.0363	0.0255	0.0331	0.0306	<u>0.0484</u>	0.0507
	NDCG@10	0.0145	0.0286	0.0319	0.0146	0.0332	0.0393	0.0455	0.0327	0.0403	0.0404	<u>0.0566</u>	0.0604
	NDCG@20	0.0193	0.0361	0.0394	0.0186	0.0410	0.0473	0.0548	0.0403	0.0485	0.0495	<u>0.0655</u>	0.0701
Toys	HR@5	0.0173	0.0438	0.0320	0.0090	0.0416	0.0565	0.0668	0.0493	0.0573	0.0579	<u>0.0766</u>	0.0832
	HR@10	0.0297	0.0642	0.0490	0.0168	0.0631	0.0800	0.0958	0.0709	0.0818	0.0872	<u>0.1032</u>	0.1129
	HR@20	0.0491	0.0918	0.0733	0.0296	0.0913	0.1089	0.1307	0.0972	0.1134	0.1210	<u>0.1348</u>	0.1498
	NDCG@5	0.0109	0.0291	0.0211	0.0054	0.0279	0.0391	0.0396	0.0336	0.0393	0.0331	<u>0.0560</u>	0.0595
	NDCG@10	0.0149	0.0357	0.0266	0.0079	0.0348	0.0467	0.0490	0.0406	0.0472	0.0426	<u>0.0646</u>	0.0690
	NDCG@20	0.0197	0.0426	0.0327	0.0111	0.0419	0.0540	0.0578	0.0472	0.0551	0.0511	<u>0.0726</u>	0.0784

addition to the conflict on item prediction task, this model modification may negatively impact the model’s effectiveness. Moreover, CL4SRec, CoSeRec, and DuoRec achieve performance improvements over other self-supervised learning-based models by incorporating contrastive learning to enhance sequence representations. CoSeRec improves CL4SRec by augmenting sequences based on item correlations and DuoRec further outperforms CoSeRec by considering semantic similarities between sequences during augmentation. These results for general sequential models and models with self-supervised learning highlight the effectiveness of the transformer encoder architecture with attention mechanisms and the advantages of introducing contrastive learning for representation learning.

For intent-based sequential models, they achieve superior performance compared to other baselines by interpreting the representations of the user interaction sequences as user intent from the user behavioral perspective. While the performance of DSSRec exceeds the performance of SASRec, ICLRec, IOCRec, and ICSRec surpass DSSRec by enhancing intent representations through contrastive learning. ICLRec and IOCRec demonstrate comparable performance with employing different contrastive learning strategy. ICLRec captures contrast signals between different intents and between intents and their associated clusters, whereas IOCRec focuses on capturing contrast signals between multiple intents. DuoRec and ICSRec consistently outperform these models and show the best performance among the baselines. Unlike other models that utilize contrastive learning with sequence augmentation which modifies the original sequence by altering item composition, they determine the sequence having the same target item with the original sequence as the augmented sequence. Additionally, they augment the training dataset itself through a splitting operation for model training. Overall, learning sequence representations by giving them the meaning of user intent from the perspective of user behavior is shown to be more effective in understanding users’ preferences compared

to learning without attributing specific semantics to sequences. Also, results indicate that sequence augmentation based on the target item can effectively generate sequences with similar context and data augmentation through sequence splitting can enrich the training process by expanding the dataset.

Finally, IDCLRec demonstrates the best performance across all baselines under all evaluation scenarios. Specifically, IDCLRec achieves 5.47% to 14.62% and 4.75% to 12.70% performance improvements over the best baseline in HR and NDCG, respectively. These results confirm the effectiveness of our proposed model. In detail, both HR and NDCG consistently exhibit highly improved performance. The improvement in HR suggests that IDCLRec can effectively identify relevant items within the recommendation lists by accurately capturing user preferences. Similarly, the improvement in NDCG implies that IDCLRec can retrieve the correct ranking of items by prioritizing more relevant items higher in the list. This consistent improvement indicates the model capability of understanding preferable items and their relative importance. Furthermore, the average performance improvement in the Sports dataset is 12.44% which is significantly higher than the 6.93% in Beauty and 8.37% in Toys. Where Sports is little sparser than other datasets, this degree of overall improvement suggests that our model can excel in handling data sparsity with capturing subtle user intent and user preferences. Regarding the development of discovered beneficial components by addressing overlooked points, we can conclude key factors of IDCLRec which contribute to overall performance improvements as follows: 1) Beyond merely considering user intents within user behaviors, our approach which explicitly explore both user interests and intents enables more comprehensive user behavior modeling. 2) The introduction of a contrastive loss function appropriate for the intent perspective with accounting for the characteristics of user intent facilitates more precise intent representation learning.

Table 4: Ablation study of the proposed model.

Model	Sports		Beauty		Toys	
	HR@20	NDCG@20	HR@20	NDCG@20	HR@20	NDCG@20
IDCLRec	0.0878	0.0426	0.1399	0.0701	0.1498	0.0784
(A) w/o intent difference minimization	0.0867	0.0425	0.1386	0.0695	0.1476	0.0776
(B) w/o intent-intent contrastive learning	0.0837	0.0407	0.1377	0.0693	0.1457	0.0766
(C) w/o intent-item contrastive learning	0.0852	0.0410	0.1361	0.0684	0.1427	0.0758
(D) w/o all auxiliary losses	0.0806	0.0379	0.1347	0.0664	0.1386	0.0712
(E) w/o intent-interest disentanglement	0.0778	0.0355	0.1325	0.0633	0.1335	0.0656
(F) w/o intent categorization	0.0799	0.0394	0.1309	0.0669	0.1401	0.0749
(G) w/o importance-weighted attention	0.0859	0.0418	0.1368	0.0691	0.1474	0.0776

5.3 Ablation Study

To investigate the effectiveness of each component of our proposed model, we conduct an ablation study on seven variants of IDCLRec in terms of training loss and model architecture. The four variants focusing on the effect of different training losses are detailed as follows: (A) removes the minimization of consecutive intent differences by setting L_d as 0, (B) removes intent-intent contrastive learning by setting L_{CL_1} as 0, (C) removes intent-item contrastive learning by setting L_{CL_2} as 0, and (D) removes all auxiliary losses by setting L_d , L_{CL_1} , and L_{CL_2} as 0. The other three variants focusing on the effect of model architecture, applying all auxiliary losses to the intents derived in each case, are as follows: (E) treats whole behaviors as only intents without disentanglement of intents and interests in behaviors by setting $H^u = I^u$, (F) uses the most recent intent instead of the categorized intent by setting $I^u = I_N^u$, and (G) uses average pooling instead of the importance-weighted attention mechanism for intent importance extraction by setting $I^u = \sum_{i=1}^{|I_{sim}^u|} \frac{1}{|I_{sim}^u|} \tilde{I}_i^u$. Table 4 reports HR@20 and NDCG@20 for the proposed model and its variants. Overall, the results clearly reveal that IDCLRec consistently outperforms all other variants, indicating the effectiveness of the proposed components in enhancing the model's ability to capture user preferences accurately.

5.3.1 Effect of Training Loss. Comparing IDCLRec with variants (A), (B), and (C), we observe that the removal of any loss function results in a decrease in performance. This implies that all the proposed loss functions are essential for improving the recommendation quality, while the importance of each depends on the degree of performance degradation. For (A), the performance decrease suggests that constraining intents to be closer in consecutive time steps helps to capture the user intents behind the users' sequential behaviors. However, the relatively small performance drop could be explained by the fact that the characteristics of intents might be implicitly contained without explicitly aligning consecutive intents. This might be because intents are expressed by subtracting the learned interest representations reflecting its properties from the behavior representations which consists of interests and intents, thereby residual intents preserve some of the underlying intent properties. The performances for (B) and (C) show that both types of contrastive learning loss functions contribute positively to express complicated intents. Specifically, (C) shows more performance degradation in Beauty and Toys while (B) is more poor in Sports. From this results, we can check that the intent enhancement through contrastive learning between intents and items is more significant in Beauty and Toys since item combinations might allow to observe specific intents although each item interaction could be occurred under multiple intents. In contrast, Sports shows

more enhancement through contrastive learning between intents, implying that each item in Sports might be strongly associated with a particular intent. By the way, in different situations, we can conclude that modeling intents in relation to item combinations or single item itself is beneficial as the model can directly connect user intents to interacted items. Furthermore, the superior performance of (A) compared to (B) and (C) suggests that integration of both proposed contrastive learning losses can have a synergistic effect in helping to express user intent more comprehensively.

In variant (D) that excludes all loss functions in model training, the performance is worse than variants which remove only one of each loss function. It can be seen that all the proposed loss functions work together in a positive direction for performance improvement without distorting the user representation. When comparing IDCLRec, (D), and ICSRec which is SOTA baseline, we observe that (D) achieves consistently higher HR than SOTA baseline, although its NDCG is lower. However, in IDCLRec that is enhanced form of (D) by training with all three proposed losses, NDCG substantially improves and ultimately reaches the best performance. A distinctive feature of (D) is its model architectural approach: 1) disentangles user behaviors into intents and interests and 2) enhances the most recent intent into the intent category it belongs to by grouping intents at each user-level. The higher HR compared to the SOTA baseline indicates that this approach is particularly effective in finding items that match user preferences. Furthermore, the three loss functions are specifically designed to capture different characteristics of intents and facilitate a nuanced adjustment of intents. The significant increase in NDCG when incorporating all proposed losses in IDCLRec validates that careful adjustment of intent representations through these losses leads to more precise ranking of preferred items, beyond merely finding them. These detailed observations on each evaluation metric demonstrate the architectural and training advantages of the proposed model in both identifying and accurately ranking items according to user preferences.

5.3.2 Effect of Model Architecture. For variant (E), the result reveals remarkably lower performance compared to IDCLRec and even (D) despite being trained with all auxiliary loss functions. This sharp performance drop highlights the importance of viewing user behaviors as combination of intents and interests to thoroughly interpret user behaviors and provide accurate recommendations. In detail, we can verify that viewing entire behaviors as intents as in existing studies leads to unclear exploration of the behaviors due to oversimplified generalizations. Where the losses are designed and applied based on the characteristics of intents, (E) fails to distinguish between intents and interests. Thus, by simplifying behaviors which inherently include both intents and interests exclusively as intents and applying intent-specific losses to them, the representations fail to disentangle distinct information, leading to entangled and ambiguous representations that degrade the overall quality of the recommendation. This observation underscores that user behaviors are not solely composed of intents but consist of both intents and interests that possess distinct characteristics that require tailored representation learning based on their unique properties.

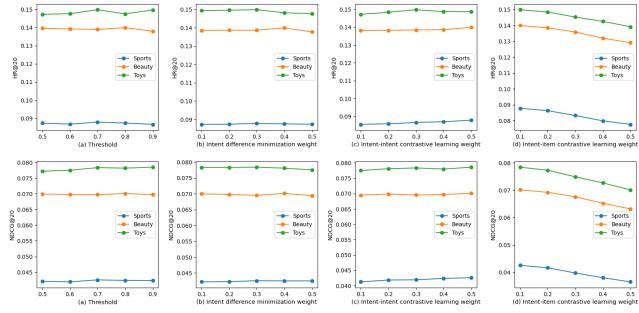


Figure 5: Performance comparisons of the proposed model on different hyperparameters.

The performance degradation from (F) suggests that reflecting only the intent for a single interaction is insufficient to robustly represent users' current purposes. Instead, aggregating similar intents across multiple interactions to learn user-wise categorical intents is more powerful for capturing user intents. Particularly, the noticeable drop in HR compared to NDCG implies that reflecting only the most recent intent provides overly strong signals confined to a specific interaction. This lack of generalizability for intentions may limit the ability to seek for diverse candidate items that align with subsequent interactions. The proposed intent categorization in IDCLRec achieves improvements by considering how categorical characteristics of intents emerge from user behaviors and adapting these characteristics from the perspective of each individual user. By doing so, it can effectively replace previous approaches that arbitrarily divide all users' intents into a fixed number of intent categories. Moreover, the impact of integrating intents about multiple interactions aligns with the observed effectiveness of learning intents through item combinations associated with similar intents as mentioned in (C). Finally, the result for (G) confirms that accounting for the varying importance of different intents within the same intent category yields a performance improvement. The higher performance of (G) compared to (F) indicates that all similar intents can collectively represent an intent category to some extent and contribute to offer a list of items that match the intention for the next interaction. However, the superior performance of IDCLRec implies that certain intents can play a more crucial role in defining their assigned intent category. Therefore, weighting each intent according to its relevance within the assigned category enables to generate more accurate intent representations.

5.4 Hyperparameter Sensitivity Analyses

Our model training involves four hyperparameters. To assess the sensitivity of each hyperparameter, we implement experiments by setting five candidate values for each. For each experiment, we change only value of target hyperparameter at a time, keeping the others fixed at their optimal values. The results for HR@20 and NDCG@20 across different hyperparameter combinations are shown in Figure 5.

5.4.1 Impact of Threshold. The threshold δ defines the similarity criterion for selecting relevant intent representations when representing categorical intent. Figure 5-(a) is results for five threshold

candidates {0.5, 0.6, 0.7, 0.8, 0.9}. The optimal threshold value is found to be 0.7 for Sports and Toys and 0.8 for Beauty. Where there are little performance differences from different value of thresholds, we can think that the results are similar across all candidates. When the threshold is lower, more intents including those with lower similarity are considered to capture the intent category, resulting in a more inclusive representation. However, through the proposed importance-weighted attention mechanism, intent representations with lower similarity to others within the same intent category are naturally assigned less weight. This allows highly similar intents to dominate the final categorical intent representation. Consequently, the final representation can be close to representation formed by only high-similarity intents selected from higher thresholds. Moreover, this understanding about the role of importance-weighted attention can implicitly explain the observed performance drops when average pooling is used instead of attention in the ablation study. To further verify this observation regarding the consistency of the results, we examine the case when the threshold is set to 0 which means all intents are considered for categorical intent learning. In this scenario, the performance is almost identical to the performance when the threshold is 0.5 with HR differing by less than 0.0007 and NDCG varying by less than 0.0001 across all datasets. These results can be explained by two factors. First, since similarity is calculated as the sigmoid of a dot product, very few intent pairs may actually have similarity below 0.5. Thus, including all intents or only those above a similarity threshold of 0.5 may yield almost the same set of intents. Second, these results also highlight the effectiveness of the importance-weighted attention that can naturally mitigate the influence of less similar intents regardless of whether a threshold is applied. Overall, while observed optimal thresholds can output the best performances, given that performances remain stable across various thresholds, introducing an additional hyperparameter that is threshold may be unnecessary. Instead, we can consider all intents for the intent importance extraction module by relying on the proposed attention mechanism to effectively emphasize the most relevant intents.

5.4.2 Impact of Intent Difference Minimization Weight. The intent difference minimization loss weight λ_d controls the degree of the loss function L_d that captures the dynamic nature of intents throughout training. We verify the impact of λ_d with five candidates {0.1, 0.2, 0.3, 0.4, 0.5} as displayed in Figure 5-(b). The best performance is achieved when weight is 0.3 for Sports and Toys and 0.4 for Beauty. In general, the variations in weight do not result in remarkable performance differences, demonstrating that the model maintains robust performance across different weight values. The mere reflection of L_d can effectively encourage similar intents for adjacent user interactions. This low sensitivity might also explain an implicit reflection of the dynamic characteristics of intents within the originally extracted intent representations.

5.4.3 Impact of Intent-Intent Contrastive Learning Weight. The intent-intent contrastive learning loss weight λ_{CL_1} controls the degree of the loss function L_{CL_1} that aligns intents for interacting same items. For five candidates {0.1, 0.2, 0.3, 0.4, 0.5}, 0.5 is optimal value for Sports and Beauty and 0.3 is optimal for Toys. As shown in Figure 5-(c), there is some sensitivity to λ_{CL_1} where overall performance improves as λ_{CL_1} increases. Specifically, for Sports and

Beauty, performance consistently improves with higher values. For Toys, while 0.3 is selected to be optimal value, 0.4 and 0.5 yield comparable results. Hence, we can conclude that higher values of λ_{CL_1} generally lead to better performance across datasets.

5.4.4 Impact of Intent-Item Contrastive Learning Weight. The intent-item contrastive learning loss weight λ_{CL_2} controls the degree of the loss function L_{CL_2} that aligns intents and corresponding item combinations. We conduct experiments with five candidates $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ and the results are illustrated in Figure 5-(d). Across all datasets, the best performances are consistently observed when the value is set to 0.1 that is the smallest value among the candidates. λ_{CL_2} exhibits high sensitivity with performance deteriorating significantly as the value increases unlike other hyperparameters.

Regarding the observed impact of each loss weight, we further interpret the results in terms of the relationship between loss weights and model performances. Related to the ablation study results, there are two interesting insights about loss weights and performances for contrastive learning tasks: 1) Although the optimal weight for intent-item contrastive learning is small as 0.1, intent-item contrastive learning loss plays a critical role in model performance by showing a significant performance drop when removing it. 2) Despite intent-intent contrastive learning requires higher optimal weights compared to intent-item, the ablation results indicate that adding intent-item contrastive learning generally leads to a larger performance improvement. Loss weights are used to balance the influence of different loss functions during model training. Although the intent-item contrastive learning loss has smaller assigned weight compared to other losses, its presence is crucial in guiding the model toward better performance. Also, a higher loss weight does not necessarily imply a greater impact on model performance. In this point, we can think that the effectiveness of each loss function depends not only on their assigned weights but also on the quality and uniqueness of the information it provides. The intent-item contrastive learning loss may offer valuable insights that other tasks cannot cover so that removing this loss can lead to a more performance degradation due to the absence of such unique characteristics. In contrast, the relatively less performance drop from the removal of the intent-intent contrastive learning loss may suggests that it provides some redundant information. From this perspective, we can think that the effectiveness of a loss function is determined more by the information it provides than by its assigned weight in this study. Thus, despite its smaller optimal weight, intent-item contrastive learning can yield a more performance improvement than intent-intent contrastive learning. We can infer that adjustment of intents through multiple item combinations can capture essential patterns of intent representations for better recommendations whereas adjustment through a single item might learn less critical information.

5.5 Intent-Interest Analysis

To investigate the distinction between intents and interests extracted from user behaviors, we visualize these two features using t-SNE [42]. As shown in Figure 6, the intent and interest representations of each user at their most recent time step are projected onto a 2D space. The resulting visualization reveals a strikingly clear separation between intent and interest representations with

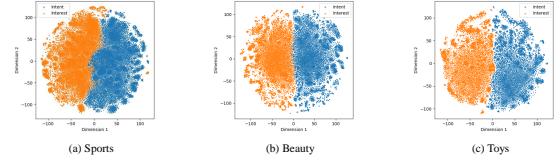


Figure 6: Visualization of intent and interest representations.

minimal overlap. This separation demonstrates that intents and interests are inherently different user characteristics. Intents, reflecting motivational short-term behaviors, are distinctly localized compared to interests which encapsulate more personal long-term behaviors. The lack of significant overlap between the two distributions highlights their complementary roles in characterizing user behaviors. Moreover, the result emphasizes the effectiveness of our proposed disentanglement approach, particularly the causal cross-attention mechanism, in capturing and isolating these distinct behavioral factors. Overall, the visualization provides compelling evidence of the distributional independence of intents and interests and supports the efficacy of our method in modeling each intent and interest considering their unique properties.

6 Conclusion

In this paper, we propose IDCLRec, a novel model that captures user behaviors more accurately by analyzing the nuanced semantics and characteristics underlying the behaviors that yield user-item interactions. Unlike previous studies that interpret user behaviors solely based on intents or without semantic frameworks, IDCLRec disentangles behaviors into intents and interests. We extract interests through a causal cross-attention mechanism that reflects the consistent characteristics of interest over time, whereas deriving intents through an additional loss function and importance-weighted attention that account for the dynamic and categorical characteristics of intent. Moreover, we introduce two types of contrastive learning loss functions that address different relationships between intents and items to better understand complex user intents. Through this modeling, IDCLRec achieves a more refined understanding of user behaviors. Experiments on real-world datasets demonstrate the effectiveness of IDCLRec through significant performance improvements.

References

- [1] Jimmy Lei Ba. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).
- [2] Russell W Belk. 1975. Situational variables and consumer behavior. *Journal of Consumer research* 2, 3 (1975), 157–164.
- [3] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. 2013. Recommender systems survey. *Knowledge-based systems* 46 (2013), 109–132.
- [4] 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*. 388–397.
- [5] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential recommendation with graph neural networks. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 378–387.
- [6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

- [7] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*. 2172–2182.
- [8] Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [9] James F Engel, David T Kollat, and Roger D Blackwell. 1969. Personality measures and market segmentation: Evidence favors interaction view. *Business Horizons* 12, 3 (1969), 61–70.
- [10] 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 *Proceedings of the ACM web conference 2022*. 2036–2047.
- [11] Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. 2020. Cert: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766* (2020).
- [12] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821* (2021).
- [13] 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*. 9729–9738.
- [14] 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)*. IEEE, 191–200.
- [15] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. 507–517.
- [16] B Hidasi. 2015. Session-based Recommendations with Recurrent Neural Networks. *arXiv preprint arXiv:1511.06939* (2015).
- [17] Dietmar Jannach and Markus Zanker. 2024. A Survey on Intent-aware Recommender Systems. *ACM Transactions on Recommender Systems* (2024).
- [18] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.
- [19] Di Jin, Luzhi Wang, Yizhen Zheng, Guojie Song, Fei Jiang, Xiang Li, Wei Lin, and Shirui Pan. 2023. Dual intent enhanced graph neural network for session-based new item recommendation. In *Proceedings of the ACM Web Conference 2023*. 684–693.
- [20] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.
- [21] Yan Kang, Yancong Yuan, Bin Pu, Yun Yang, Lei Zhao, and Jing Guo. 2024. HICL: Hierarchical Intent Contrastive Learning for sequential recommendation. *Expert Systems with Applications* 251 (2024), 123886.
- [22] Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [23] Walid Krichene and Steffen Rendle. 2020. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 1748–1757.
- [24] Haoyang Li, Xin Wang, Ziwei Zhang, Jianxin Ma, Peng Cui, and Wenwu Zhu. 2021. Intention-aware sequential recommendation with structured intent transition. *IEEE Transactions on Knowledge and Data Engineering* 34, 11 (2021), 5403–5414.
- [25] Jiacheng Li, Tong Zhao, Jin Li, Jim Chan, Christos Faloutsos, George Karypis, Soo-Min Pantel, and Julian McAuley. 2022. Coarse-to-fine sparse sequential recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*. 2082–2086.
- [26] Xuwei Li, Aitong Sun, Mankun Zhao, Jian Yu, Kun Zhu, Di Jin, Mei Yu, and Ruiqiu Yu. 2023. Multi-intention oriented contrastive learning for sequential recommendation. In *Proceedings of the sixteenth ACM international conference on web search and data mining*. 411–419.
- [27] Yue Liu, Shihao Zhu, Jun Xia, Yingwei Ma, Jian Ma, Wenliang Zhong, Xinwang Liu, Guannan Zhang, and Kejun Zhang. 2024. End-to-end learnable clustering for intent learning in recommendation. *arXiv preprint arXiv:2401.05975* (2024).
- [28] 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).
- [29] Jie Lu, Dianshuang Wu, Mingsong Mao, Wei Wang, and Guangquan Zhang. 2015. Recommender system application developments: a survey. *Decision support systems* 74 (2015), 12–32.
- [30] Jianxin Ma, Chang Zhou, Hongxiao 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*. 483–491.
- [31] 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*. 43–52.
- [32] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).
- [33] Zhiqiang Pan, Fei Cai, Yanxiang Ling, and Maarten de Rijke. 2020. Rethinking item importance in session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 1837–1840.
- [34] 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*. 89–98.
- [35] Xiuyuan Qin, Huanhuan Yuan, Pengpeng Zhao, Guanfeng Liu, Fuzhen Zhuang, and Victor S Sheng. 2024. Intent Contrastive Learning with Cross Subsequences for Sequential Recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 548–556.
- [36] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*. 813–823.
- [37] 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*. 1441–1450.
- [38] Qiaoyu Tan, Jianwei Zhang, Jiangchao Yao, Ninghao Liu, Jingren Zhou, Hongxia Yang, and Xia Hu. 2021. Sparse-interest network for sequential recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining*. 598–606.
- [39] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 17–22.
- [40] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 565–573.
- [41] Md Mehrab Tanjim, Congzhu Su, Ethan Benjamin, Diane Hu, Liangjie Hong, and Julian McAuley. 2020. Attentive sequential models of latent intent for next item recommendation. In *Proceedings of The Web Conference 2020*. 2528–2534.
- [42] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).
- [43] A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).
- [44] Lei Wang, Ee-Peng Lim, Zhiwei Liu, and Tianxiang Zhao. 2022. Explanation guided contrastive learning for sequential recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2017–2027.
- [45] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z Sheng, and Mehmet Orgun. 2019. Sequential recommender systems: challenges, progress and prospects. *arXiv preprint arXiv:2001.04830* (2019).
- [46] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.
- [47] Yuling Wang, Xiao Wang, Xiangzhou Huang, Yanhua Yu, Haoyang Li, Mengdi Zhang, Zirui Guo, and Wei Wu. 2024. Intent-aware recommendation via disentangled graph contrastive learning. *arXiv preprint arXiv:2403.03714* (2024).
- [48] 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)*. IEEE, 1259–1273.
- [49] 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*. 3434–3445.
- [50] 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*. 1893–1902.
- [51] Peilin Zhou, Jingqi Gao, Yueqi Xie, Qichen Ye, Yining Hua, Jaeboun Kim, Shoujin Wang, and Sungjun Kim. 2023. Equivariant contrastive learning for sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 129–140.
- [52] Peilin Zhou, You-Liang Huang, Yueqi Xie, Jingqi Gao, Shoujin Wang, Jae Boum Kim, and Sungjun Kim. 2024. Is Contrastive Learning Necessary? A Study of Data Augmentation vs Contrastive Learning in Sequential Recommendation. In *Proceedings of the ACM on Web Conference 2024*. 3854–3863.