Information Processing and Management 61 (2024) 103619

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Disentangle interest trend and diversity for sequential recommendation

Zihao Li [a], Yunfan Xie [a], Wei Emma Zhang [b], Pengfei Wang [c], Lixin Zou [a], Fei Li [a], Xiangyang Luo [d], Chenliang Li [a,*]

[a] *Key Laboratory of Aerospace Information Security and Trusted Computing, School of Cyber Science and Engineering, Wuhan University, Wuhan, 430072, Hubei, China*
[b] *School of Computer and Mathematical Sciences, The University of Adelaide, Adelaide, 5005, Australia*
[c] *School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, 100876, China*
[d] *PLA Strategic Support Force Information Engineering University, Zhengzhou, 450001, Henan, China*

ARTICLE INFO

Keywords:
Sequential recommendation
Disentangled representation
Multi-interest learning

ABSTRACT

Based on historical behaviors, sequential recommendation endeavors to predict what a user prefers next. The recent efforts are mainly devoted to modeling the user's interests evolution process or mining multi-interests for recommendation. However, it is largely overlooked that the *interest trend* (i.e., the evolution of the main interest) and the *interest diversity* (i.e., the scattered potential interests) could complement each other for better performance. Specifically, the interest trend reveals the user's basic interest and its evolution, which is satisfied by similarity recommendations. Nevertheless, interest diversity covers the various interests caused by some external environmental influence, e.g., fashion trends and advertisements, exploring users' potential interests or interest diversity will facilitate the model for diversity and serendipity recommendation. In a way, these two factors have conflicting aims, we argue that they should be disentangled in modeling first and recombined when making personalized recommendation.

To instantiate this idea, we propose a simple yet effective model, dubbed Teddy (disentangles interest **trend** and **d**iversity), which disentangles and then jointly models the aforementioned two factors under a unified framework. Particularly, in Teddy, an adaptive masking mechanism is first introduced to split the user's historical items into two parts revealing her major interest trend and scattered interest diversity respectively for interests disentanglement. Then, a temporal convolutional network (TCN) is utilized to capture the evolution process of the user's major interest trend. For the scattered interest diversity modeling, we further choose to apply Multilayer Perceptron (MLP) layers with max-pooling mechanism to extract the significant or dominant preference signals. The predicted scores generated by these two modules are aggregated together to integrate both *interest trend* and *interest diversity* for the final recommendation. Extensive experiments over four public datasets demonstrate the superiority of our proposed Teddy against a series of SOTA alternatives on the benchmark metrics.

1. Introduction

Recommendation systems have become an indispensable installation in multiple online services, e.g., electronic commerce, entertainment, and social media applications. Aiming to capture the dynamic user preference based on his or her rich historical

* Corresponding author.
E-mail address: cllee@whu.edu.cn (C. Li).

https://doi.org/10.1016/j.ipm.2023.103619
Received 28 August 2023; Received in revised form 13 November 2023; Accepted 23 December 2023
Available online 4 January 2024

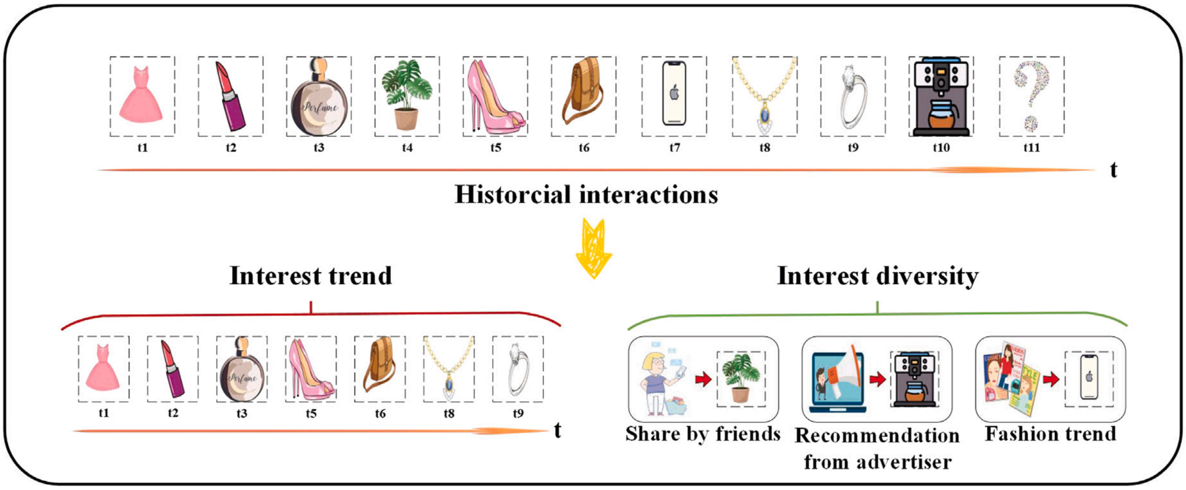

Fig. 1. The user historical interactions can be split into two parts: (1) major interest trend (red bracket); (2) interest diversity (green bracket). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

interaction records, such as view, purchase, add to chat, comments, etc., for next item prediction (Wang et al., 2019), sequential recommendation has drawn great attention from both academia as well as industry very recently.

Earlier efforts mainly focus on modeling the user's interest evolution with sequential neural networks and achieving remarkable success (Hidasi & Karatzoglou, 2018; Kang & McAuley, 2018; Sun et al., 2019; Tang & Wang, 2018). For instance, many methods leverage the sequential modeling ability of RNN or CNN to approximate the user's next interest (Hidasi & Karatzoglou, 2018; Tang & Wang, 2018; Yakhchi et al., 2022). Also, self-attention mechanism is adopted to aggregate the relevant signals over the historical sequence for target item prediction (Kang & McAuley, 2018; Sun et al., 2019). Furthermore, some sophisticated techniques, e.g., memory neural network (Chen et al., 2018; Tan et al., 2021), knowledge graph (Huang et al., 2018; Wang et al., 2021), contrastive learning (Chen, Liu et al., 2022; Du et al., 2022; Ni et al., 2023), reinforcement learning (Tong et al., 2021), are utilized to enhance interest extraction for better performance. Recently, Graph Neural Networks (GNNs) have shown great effectiveness by aggregating the high-order neighborhood information (Fan, Liu, Zhang et al., 2021; Guo et al., 2021). Although these works deliver significant improvement for sequential recommendation, they mainly specialize in the interest evolution process or major interests modeling (i.e., *interest trend*), which are highly correlated with the user's profile like social status, financial power, education level, and occupation, to name a few; while neglecting the extraordinarily diverse and dynamic property of the user's interests (i.e., *interest diversity*), leading models to achieve sub-optimal performance.

In recent years, some works propose multi-interest modeling (Cen et al., 2020; Li et al., 2019; Pi et al., 2019; Tian et al., 2022; Zhang, Yang et al., 2022) or perform denoising (Lin et al., 2022; Tong et al., 2021) for sequential recommendation. However, given a user's historical sequential interactions, those works either treat multiple interests equally or believe those diverse items inconsistent with the user's basic interest are noise and should be removed, hindering the learning of the user's subtle preference. Here, we argue that these two treatments bear some drawbacks respectively.

- **Modeling interest trend and diversity together lead to performance degradation.** Note that the user's interests could be decomposed into the combination of (1) interest trend; and (2) interest diversity. In particular, the interest trend reveals the user's long-term interest evolution process that will not shift sharply as time goes by and encourages *similar recommendations*, while interest diversity, tends to explore a user's various point-of-interests and prompt more *diverse recommendations*. Therefore, modeling the objective-conflict factors and optimizing the orthogonal aims, i.e., the interest trend and interest diversity, simultaneously without disentanglement will lead to performance deterioration.
- **Eliminating the diverse items results in significant information loss**. To avoid the above performance degradation, existing methods treat the interest diversity as noise (Fan et al., 2022; Sun et al., 2021; Wang, Zhang et al., 2022) and remove the diverse and irrelevant items sloppily. However, we argue these items could be attributed to external environmental factors, e.g., emerging fashion trends, advertisement and marketing, and sharing or recommendation from friends, thus, they may also represent the user's potential interests that could facilitate the point-of-interest exploration.

Fig. 1 illustrates such an example, where the user is a well-educated young female with a substantial income. Her historical interactions (i.e., the major interest trend) mainly focus on cosmetics, jewels, luxurious clothing and bags. However, some delicate items, e.g., fresh flowers and potted plants, coffee machine, and the latest model of the iPhone, which may be recommended by friends or advertisers are also her scattered interests. Therefore, we believe it may be more appropriate to consider those items as user's interest diversity. It is noteworthy that these two parts could reveal a user's different aspects of interests and both of them are equally important for recommendation. Unfortunately, none of the existing works disentangles such two kinds of interests first and model them respectively under a unified framework.

To bridge this gap, we propose a simple yet effective model that disentangles interest **trend** and **d**iversit**y**, named Teddy, for effective sequential recommendation. Specifically, we obtain the user's recent interactions which could reveal the user's current preference and also the major interests and Based on an adaptive masking module, Teddy firstly extracts a user's current intention to generate a mask vector, which is further utilized to splits the user's historical items into two parts, corresponding to the major interest trend and diversity respectively. Then, a temporal convolutional neural network (TCN) is introduced to model the evolution of the major interest trend. As to the scattered interest diversity, we choose to derive the user representation with MLP and max-pooling mechanism. Afterward, the final score is generated by aggregating these two factors together for recommendation. It is noteworthy that the user's interest trend or major interests which often consistent with the user's current preference and dominate the final recommendation results, while the interest diversity can be recognized as the potential interests and the supplementary to the interest trend recommendation results (ref. Table 3 for detail). Consequently, we believe modeling user's interest trend as well as interest diversity simultaneously will provide a more comprehensive consideration for sequential recommendation. In a nutshell, our contributions can be summarized as below:

- We devise a novel sequential recommendation method, dubbed Teddy. To the best of our knowledge, this is the first attempt to model a user's major interest trend and scattered interest diversity in a holistic approach for an effective sequential recommendation.
- We propose an adaptive masking mechanism to split a user's historical sequence into two segments with regard to the major interest trend and scattered interest diversity respectively. Moreover, a TCN module and an MLP with max-pooling module are dedicated, which allows us to capture the above two types of interests with tailored strategies separately.
- Our extensive experiments on four public datasets demonstrate that the proposed Teddy significantly outperforms a series of baselines and state-of-the-art solutions. Further analysis is also conducted to validate each design choice and the rationality of our model.

The rest of the paper is structured as follows. Sections Section 2 reviews related works in this relevant research direction. Section 3 introduces the implementation of our proposed Teddy and each module in detail. The experiment results and in-depth analysis are presented in Section 4. In the end, we conclude this work and further discuss future research directions in Section 5.

2. Related work

In this section, we first introduce some representative methods including Markov Chain and deep learning method for sequential recommendation briefly. As sequential recommendation with multi-interest modeling and denoising are highly related to our work, we further give a comprehensive summary of typical models that have emerged in these two research lines.

2.1. Markov chain-based methods

Conventional sequential recommendation methods focus on exploiting the item transition patterns (Rendle et al., 2010; Shani et al., 2005; Zimdars et al., 2013). For instance, Markov Chain models the behavior of next interaction as a Markov Decision Process (MDP) and learning the state transfer matrix for the next item recommendation. To be specific, Shani et al. (2005) and Zimdars et al. (2013) endeavor to devise different Markov models for sequential pattern extraction and next item prediction. Rendle et al. (2010) propose Factorizing Personalized Markov Chain (FPMC) to model the adjacent interaction behaviors via the factorization of users' personalized probability transition matrices. Although, as a prominent solution in the early stage, Markov Chain models achieve promising success in sequential recommendation, the strict first-order (i.e., the next interacted item only decided by the previous one) hinders the model to capture more complicated user behaviors and also constrained the development of this method.

2.2. Sequential neural networks

Recently, a plethora of methods apply neural networks to capture the implicit patterns encapsulated in the sequences, e.g., LSTM, GRU, and CNN (Kang & McAuley, 2018; Sun et al., 2019; Tang & Wang, 2018). For instance, GRU4Rec (Hidasi et al., 2015) is the first that attempted to leverage GRU for sequential recommendation. Specifically, it stacks multiple GRU layers and uses a ranking loss function for performance improvements. HRNN (Quadrana et al., 2017) develops hierarchical RNNs, which introduce the user's whole historical interacted records as auxiliary information for a more precise personalized recommendation. As a follow-up research, Xiao et al. (2019) propose a hierarchical neural variational model (HNVM) to enhance the capability of vanilla GRU via a hierarchical Gaussian latent representation. Such that, HNVM could capture the users' long-term and short-term interests in a more flexible and sophisticated fashion. To better capture the significance of each interacted item to the current target item, the attention mechanism is also adopted for sequential recommendation. More concretely, as a pioneer work, SASRec (Kang & McAuley, 2018) uses a unidirectional self-attention to obtain the implicit connection between each historically interacted item and the current target item for recommendation. In contrast, BERT4Rec (Sun et al., 2019) applies a bidirectional self-attention with a Cloze task (Taylor, 1953) to capture both user's previous and future information for the current preference prediction. SSE-PT (Wu et al., 2020) concates user embedding with target item embedding for personalized sequential recommendation. Furthermore, by defining convolution filters, a convolution neural network is also utilized to obtain both general preferences and local patterns (Tang & Wang, 2018; Ye et al., 2020) for sequential recommendation. Besides, some well-design modules, e.g., memory neural network (Chen et al., 2018; Tan et al., 2021), are adopted to improve the performance of sequential recommendation. Despite the encouraging performance obtained by these sequential models, they only explore the sequential information from the same sequence, while leaving out the high-order correlations to be unexploited explicitly, which are also important for sequential recommendation. To tackle these problems, a surge of GNNs is proposed. Attribute to the information propagation and aggregation and collaborative sequence mining, GNN models have achieved competitive performance on sequential recommendation (Ding et al., 2021; Fan, Liu, Zhang et al., 2021).

2.3. *Graph neural network*

Endowed with the property of multi-hop contextual information prorogation and aggregation, Graph neural networks (GNNs), a more flexible solution for item transition modeling, become ubiquitous in sequential recommendation. Wu et al. (2019) propose SR-GNN, which is a pioneering work that applies GNN to capture explicit correlations between items for the next-item prediction. Considering the implicit connections between items and the changeable user preference in the sequential recommendation scenario, SURGE transforms the sparse item sequence into a tight item–item graph to extract user's core interests for recommendation (Chang et al., 2021). Additionally, Fan, Liu, Zhang et al. (2021) propose a temporal graph, which aims to model time interval information and collaborate filter signal with a holistic method for the sequential recommendation. Zhang, Wu et al. (2022) carry out a dynamic graph, which leverages item–item transition graph as well as user–item interaction graph to capture the transitional information and the collaborative filtering signals for embedding enhancement.

2.4. *Multi-interest modeling*

Encoding multifaceted interests into a single vector might be insufficient for user's preference modeling. Hence, many works are devoted to representing the user's preference in terms of multiple interest vectors. This line of efforts can be categorized into two main mainstreams: *modeling user's short- and long-term interest jointly* and *extracting multiple interests*. Often, the user's current preferences are utilized to represent her short-term or current intentions. For example, STAMP (Liu et al., 2018) utilizes a short-term attention and memory priority model to learn user's short and short-term interests simultaneously for the next-item prediction. HGN (Ma et al., 2019) utilizes item-level gating and feature-level gating mechanisms for short-term and long-term preference modeling. GLS-GLR (Wang et al., 2020) devises a group-aware short-term and long-term graph learning solution for the group recommendation.

Since user interests are diverse and uncertain, multi-interest modeling mainly utilizes dynamic routing or soft-attention recipes to represent user interests with multiple vectors. To be specific, MIMN (Pi et al., 2019) utilizes a memory neural network to extract a user's multi-interest for long sequence recommendation. MIND (Li et al., 2019) and DIN (Zhou et al., 2018) utilize soft-attention and dynamic routing respectively for multi-interest modeling. More recent works that also perform multi-interest modeling include ComiRec (Cen et al., 2020), and SPLIT (Shao et al., 2022). Disen-GNN (Li et al., 2022) casts the item embedding into multiple chunks and utilizes the graph neural network for the item's multi-factors and user's multi-interest modeling. Moreover, Fan et al. (2022) and Fan, Liu, Wang et al. (2021) believe user's sequential behaviors are uncertain in practice. Thus, a dedicated stochastic self-attention (STOSA) module is proposed to embed each item as a stochastic Gaussian distribution for dynamic interest modeling. DMRL (Liu et al., 2022) argues that different modality information can reflect different features of items and also reveal the user's modality preference. Therefore, the authors disentangle the item's multimodal representation, estimating the effects of each factor on the final recommendation in an integrative manner. Some other works like TiMiRec (Wang, Wang et al., 2022) and Re4 (Zhang, Yang et al., 2022) strive to devise auxiliary loss functions for multi-interest extraction and discrimination. However, all these methods either treat each interest equally or specialize in the short-term (i.e., the current intentions) and long-term interest (i.e., the historical preference) modeling, none of them devoted to modeling different interests as the major interest trend and the scattered interest diversity with a holistic approach.

2.5. *Denoising for sequential recommendation*

Besides extracting multiple interests, another line of effort resorts to filtering out scattered interests as irrelevant or noisy items from the historical sequence. Specifically, these denoising models can be divided into two mainstreams: implicit denoising (Yuan et al., 2021; Zhou et al., 2022) and explicit denoising (Zhang, Du et al., 2022).

Attention-based methods leverage different attention modules to model the significance of each historical item (Yuan et al., 2021) for implicit denoising. For instance, DSAN (Dual Sparse Attention Neural Network) adopts an adaptively sparse transformation operation to diminish the detriment of unrelated interactions (Yuan et al., 2021). FMLP (Filter-enhanced MLP) introduces fast Fourier transform (FFT) into the MLP layer (Zhou et al., 2022). Through different filter kernels, FMLP could eliminate the irrelevant signal from the frequency domain and capture the global dependency of sequence for recommendation. BERD (Sun et al., 2021) models the uncertainty of each individual through a Gaussian distribution. Thus, the unreliable items could be removed for a better recommendation.

In contrast, explicit denoising solutions aim to obtain the correlation between the sequence representation and target items' for noise elimination and recommendation. Zhang, Du et al. (2022) propose a hierarchical sequence denoising method (HSD), which calculates the relevance between user preference and items via a soft-attention for user-level denoising. Jin et al. (2022) apply soft and hard search strategies to eliminate irrelevant items and only keep relevant items for recommendation. Tong et al.

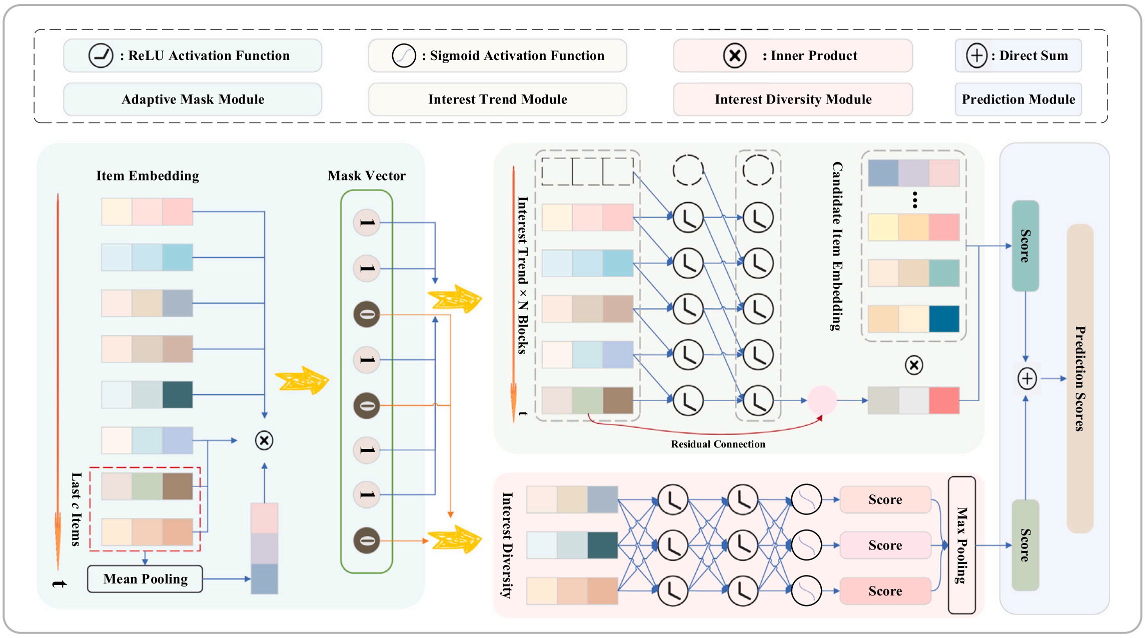

Fig. 2. The network architecture of our proposed Teddy.

(2021) formalize sequence denoising as a Markov Decision Process (MDP), then applies reinforcement learning to remove irrelevant items for recommendation. DETAIN (decomposed item feature routing algorithm) is also a similar work in the same period (Lin et al., 2022). Despite the impressive performance gain achieved by removing those irrelevant items, we believe that capturing the uncertainty is also critical to enhance sequential recommendation (Fan et al., 2022; Sun et al., 2021; Wang, Zhang et al., 2022). Hence, the proposed Teddy is devised to disentangle the user's major interest trend and scattered interest diversity respectively for sequential recommendation.

3. Methodology

In this section, we will illustrate our Teddy in detail. Specifically, we first describe the problem formulation for sequential recommendation. Afterwards, the overview of our model is presented which is then followed by the technical detail of each component.

3.1. Problem formulation

Let symbol $\mathcal{I}$ denotes the item collection for recommendation, and N denotes the collection size (i.e., $N = |\mathcal{I}|$), The historical interaction sequence for user u can be chronologically organized as $\mathcal{S}_u = \{x_{i_1}, x_{i_2}, \dots, x_{i_{l-1}}, x_{i_l}\}$ where x_{i_k} and x_{i_l} denote kth and latest item interacted by the user respectively from collection $\mathcal{I}$. The task of sequential recommendation aims to yield a ranking list of candidates as the prediction results that will more likely be clicked by user at $(l+1)$-th time step.

3.2. Model overview

As demonstrated in Fig. 2, the proposed Teddy framework contains four major components: *adaptive masking module*, *interest trend modeling module*, *interest diversity modeling module* and *prediction aggregation module*. Firstly, we utilize the adaptive masking mechanism to split the user's historical sequence $\mathcal{S}_u$ into two parts: (1) a sub-sequence $\mathcal{S}_u^m$ with regard to the use's major interest trend; (2) and an interest diversity set $\mathcal{D}_u$ containing the rest individual items with respect to the user's scattered interest diversity. After that, the major interest trend and its evolution are modeled via a temporal convolutional network (TCN). The resultant last item representation yield by TCN is obtained as the latent feature representation of $\mathcal{S}_u^m$. Let the item embedding table for all items in $\mathcal{I}$ be $\mathbf{X} \in \mathbb{R}^{N \times d}$, where d is the embedding vector dimension. The corresponding next item likelihood $\hat{\mathbf{y}}_u^m$ can be derived accordingly. In terms of the scattered interest diversity, we treat each item in $\mathcal{D}_u$ independently and utilize a two-layer MLP with the max-pooling operation to derive the corresponding next item likelihood $\hat{\mathbf{y}}_u^d$. The prediction aggregation module then combines the predictions from the two perspectives (i.e., interest trend and interest diversity) for final likelihood estimation. Note that no item will be contained by $\mathcal{S}_u^m$ as well as $\mathcal{D}_u$ simultaneously. In addition, $\mathcal{S}_u^m$ and $\mathcal{D}_u$ will be optimized by the correspondent modules with different likelihood $\hat{\mathbf{y}}_u^m$ and $\hat{\mathbf{y}}_u^d$ independently. Consequently, we believe the conflicting factors and objections from the two aspects can be disentangled and reconciled in an effective manner. To sum up, the whole process can be formalized as follows:

$$\begin{aligned} &\mathcal{S}_u \longrightarrow \{\mathcal{S}_u^m,\ \mathcal{D}_u\} \\ &\hat{\mathbf{y}}_u^m = \mathrm{trd}(\mathcal{S}_u^m, \mathbf{X}) \\ &\hat{\mathbf{y}}_u^d = \mathrm{div}(\mathcal{D}_u) \\ &\hat{\mathbf{y}}_u = f(\hat{\mathbf{y}}_u^m, \hat{\mathbf{y}}_u^d) \end{aligned} \tag{1}$$

where $\hat{\mathbf{y}}_u$ is the final likelihood estimation vector of user u generated by the prediction aggregation function $f(\cdot)$. In addition, $\mathrm{trd}(\cdot)$ and $\mathrm{div}(\cdot)$ are the interest trend modeling module and the interest diversity modeling module respectively.

3.3. Adaptive masking module

The adaptive masking module aims to split the user's historical sequence $\mathcal{S}_u$ into two parts regarding to the interest trend and interest diversity respectively, i.e., $\mathcal{S}_u^m$ and $\mathcal{D}_u$. Following the existing works (Liu et al., 2018; Luo et al., 2020), we suppose the latest c items could reflect the user's very recent preference and intentions. Consequently, we perform a linear projection for the averaged embedding of these c items in $\mathcal{S}_u$ as the proxy representation of a user's current interest, which will be applied as a supervised signal for interest trend and interest diversity split. It is intuitive that the items with regard to the user's major interest trend hold relatively higher semantic relevance against each other, while the items reflecting the scattered interest diversity could be less relevant to each other. Furthermore, the volumes of these two parts are also likely to be uneven, i.e., the number of items covering the major interest trend could largely surpass the latter in most cases (i.e., $|\mathcal{S}_u^m| > |\mathcal{D}_u|$).

Based on these two essentials, we choose to calculate the relevance between each item embedding and the proxy representation. The resultant relevance score can be utilized to generate a masking variable m_j for item i_j, reaching the purpose of disentangling the major interest trend and interest diversity. Formally, the adaptive masking mechanism works as follows:

$$\begin{aligned} &\mathbf{p} = \frac{1}{c}\sum_{j=l-c+1}^{l} \mathbf{W_s}\mathbf{x}_{i_j} \\ &r_j = \mathrm{cosine}(\mathbf{x}_j, \mathbf{p}) = \frac{\mathbf{x}_{i_j}^T \cdot \mathbf{p}}{\left\|\mathbf{x}_{i_j}\right\| \|\mathbf{p}\|} \\ &m_j = \begin{cases} 1 & , if\ r_j \geq \theta_m \\ 0 & , \text{otherwise} \end{cases} \end{aligned} \tag{2}$$

where $\mathbf{x}_{i_j} \in \mathbb{R}^d$ is the embedding of item i_j in $\mathcal{S}_u$, $\mathbf{W_s} \in \mathbb{R}^{d\times d}$ is learnable parameters, $\mathbf{p}$ is the proxy representation of the user's major interest trend, the function $cosine(\mathbf{a}, \mathbf{b})$ calculates the cosine similarity between vector $\mathbf{a}$ and $\mathbf{b}$. θ_m is a pre-defined threshold, which controls the sharpness of the interest discrimination. Here, $m_j \in \{0, 1\}$ works as the classification label of item i_j: when $m_j = 1$ then x_{i_j} is included in $\mathcal{S}_u^m$ (i.e., major interest trend); otherwise, x_{i_j} is included in $\mathcal{D}_u$ (i.e., interest diversity). When we decrease the value of θ_m, more items would be classified as the elements of the major interest trend and vice versa. Note that the interest trend sequence $\mathcal{S}_u^m$ is also organized chronologically, in order to keep the same order as in $\mathcal{S}_u$: $\mathcal{S}_u^m = \{x_{i_1^m}, x_{i_2^m}, \ldots, x_{i_{l'}^m}\}$ where $x_{i_j^m}$ was interacted by the user before $x_{i_{j+1}^m}$ but after $x_{i_{j-1}^m}$, and l' is the length of $\mathcal{S}_u^m$.

3.4. Interest trend module

To extract the latent features towards the evolution process of user's major interest trend, we choose to utilize a temporal convolution network (TCN). It has been validated that TCN is more effective for sequence modeling because the backpropagation path is in the direction of the network depth, which could alleviate the gradient exploding or vanishing problem in RNNs (Bai et al., 2018). More specifically, the TCN utilizes dilated convolutions capitalized on an exponentially large receptive field (Zhang et al., 2016) to model the long-range dependencies. Given interest trend sequence $\mathcal{S}_u^m = \{x_{i_1^m}, x_{i_2^m}, \ldots, x_{i_{l'}^m}\}$, a TCN block firstly perform a two-layer dilated convolution $\mathcal{F}$ on each element $x_{i_j^m}$ of the sequence as follows:

$$\mathbf{f}_j^1 = \mathrm{ReLU}(\mathbf{W}_t^1 \cdot [\mathbf{x}_j, \mathbf{x}_{j-w}, \ldots, \mathbf{x}_{j-w\cdot(k-1)}] + \mathbf{b}_t^1) \tag{3}$$

$$\mathbf{f}_j^2 = \mathrm{ReLU}(\mathbf{W}_t^2 \cdot [\mathbf{f}_j^1, \mathbf{f}_{j-w}^1, \ldots, \mathbf{f}_{j-w\cdot(k-1)}^1] + \mathbf{b}_t^2) \tag{4}$$

where $\mathbf{W}_t^1, \mathbf{W}_t^2 \in \mathbb{R}^{d\times kd}$ and $\mathbf{b}_t^1, \mathbf{b}_t^2 \in \mathbb{R}^d$ are learnable parameters, w and k are the dilation factor and filter size respectively. We utilize layer normalization after the dilated convolution, $\mathbf{x}_j$ is the embedding of item $x_{i_j^m}$ and we apply rectified linear unit (ReLU) as the activation function. Hence, the dilated convolution reduces to a standard 1-D convolution when w is set to be 1. In contrast, when we increase the value of w, the sequential patterns in a broader range (i.e., a wider receptive field) can be captured. Afterward, a residual connection is utilized to facilitate stacking multiple TCN blocks as follows:

$$\mathbf{o}_j = \mathrm{ReLU}(\mathbf{x}_j + \mathbf{f}_j^2) \tag{5}$$

where $\mathbf{o}_j$ is considered as the output of a single TCN block. To endow Teddy with the capacity of modeling the higher-order and more complicated sequential patterns, we choose to stack multiple TCN blocks, which can be formalized as follows:

$$\mathbf{o}_j^1 = \mathrm{TCN}(\mathbf{x}_j),\ \mathbf{o}_j^2 = \mathrm{TCN}(\mathbf{o}_j^1),\ \ldots,\ \mathbf{o}_j^{L-1} = \mathrm{TCN}(\mathbf{o}_j^{L-2}),\ \mathbf{o}_j^L = \mathrm{TCN}(\mathbf{o}_j^{L-1}) \tag{6}$$

where L is the number of layers. Following the common setting in previous works (Bai et al., 2018; Lea et al., 2017), we increase the dilation factor w exponentially with the depth of the network (i.e., $w = 2^j$ at jth layer of the stacked TCN), and a spatial dropout is added on top of each dilated convolution. In this sense, TCN is able to harness both the specific information of each individual in the bottom layers and the complex sequential patterns in the top layers.

At last, we denote the representation $\mathbf{o}_j^L$ of item $i_{l'}^m$ generated by the L-layer TCN as interest trend representation $\mathbf{d}_u$. Then, the likelihood of being the next item $\hat{\mathbf{y}}_u^m$ for user u can be calculated via the inner product between item embedding table $\mathbf{X}$ and $\mathbf{d}_u$ as follows:

$$\hat{\mathbf{y}}_u^m = \mathbf{X}\mathbf{d}_u \tag{7}$$

where jth element in $\hat{\mathbf{y}}_u^m$ is the likelihood estimated for item x_{i_j}.

3.5. Interest diversity module

As to the user's scattered interest diversity, the items included in set $\mathcal{D}_u$ would be very different. Hence, it becomes natural to identify the maximum likelihood from this set. Specifically, we extract the high-level semantic features for each item x_j in $\mathcal{D}_u$ as follows:

$$\mathbf{q}_j = \mathrm{ReLU}(\mathrm{ReLU}(\mathbf{x}_j\mathbf{W}_d^1 + \mathbf{b}_d^1)\mathbf{W}_d^2 + \mathbf{b}_d^2) \tag{8}$$

where $\mathbf{W_1}, \mathbf{W_2} \in \mathbb{R}^{d\times d}$ and $\mathbf{b_1}, \mathbf{b_2} \in \mathbb{R}^d$ are learnable parameters. $\mathbf{q}$ is the latent features extracted via a two-layer MLP. Afterward, we perform a linear regression with a max-pooling mechanism to derive the interest-diversity-based likelihood $\hat{\mathbf{y}}_\mathbf{u}^\mathbf{d}$ as follows:

$$\hat{\mathbf{y}}_\mathbf{u}^\mathbf{d} = \max(\sigma(\mathbf{Q}_u\mathbf{W}_d^3 + \mathbf{b}_d^3)) \tag{9}$$

where σ is the sigmoid activation function, $\mathbf{Q}_u = [\mathbf{q}_1; \cdots; \mathbf{q}_{n-1}; \mathbf{q}_n]$ and n is the size of $\mathcal{D}_u$, i.e., the total number of items classified as interest diversity. $\mathbf{W}_d^3 \in \mathbb{R}^{d\times N}$ and $\mathbf{b}_d^3 \in \mathbb{R}^N$ are learnable parameters. Function $max(\cdot)$ returns the maximum value for each row of the given matrix.

3.6. Prediction aggregation and optimization

Prediction Aggregation. We, thereby, first utilize a simple linear interpolation function to combine the prediction scores ($\hat{\mathbf{y}}_u^m$ and $\hat{\mathbf{y}}_u^d$) from both the interest trend modeling module and the interest diversity modeling module. It is formalized as follows:

$$\hat{\mathbf{y}} = \mathrm{softmax}(\alpha * \hat{\mathbf{y}}_u^m + (1-\alpha) * \hat{\mathbf{y}}_u^d) \tag{10}$$

In Eq. (10), we calculate the weighted sum of two predictions $\hat{\mathbf{y}}_u^m$ and $\hat{\mathbf{y}}_u^d$ with a coupling factor $\alpha \in [0,1]$. $\hat{\mathbf{y}}$ is the final likelihood score for each item.

Model Optimization. As for model optimization, we choose to learn the parameters for a given interaction sequence $\mathcal{S}_u$ by using the cross-entropy loss:

$$\mathcal{L}(\hat{\mathbf{y}}) = -\sum_{j=1}^{N} \mathbf{y}[j]\log(\hat{\mathbf{y}}[j]) + (1-\mathbf{y}[j])\log(1-\hat{\mathbf{y}}[j]) \tag{11}$$

where $\mathbf{y}$ is the ground-truth one-hot encoding vector and $\mathbf{y}[j]$ is jth element of $\mathbf{y}$, the same goes for $\hat{\mathbf{y}}[j]$.

4. Experiments

In this section, we select four open datasets and conduct extensive experiments to evaluate our proposed Teddy[1] against a series of baselines and state-of-the-art solutions e.g., representative sequential recommendation methods, multi-interests modeling solution and denoising works. First, we illustrate the experiment settings including the open datasets, baselines in comparison, evaluation metrics, experimental and hyperparameters setup in brief. Then, we discuss the performance comparison and results from the model analysis. More concretely, we aim to answer the following questions:

- **RQ1.** How does our proposed Teddy perform against other baselines for sequential recommendation?
- **RQ2.** How does each module in Teddy impact the final recommendation performance?
- **RQ3.** What is the performance of Teddy under different hyper-parameter settings?
- **RQ4.** How does Teddy explore and exploit the user's interest trend and interest diversity for performance improvement?

[1] The code implementation will be released publicly upon paper acceptance.

Table 1
Statistics of four preprocessed datasets. Avg_len: the average length of an interaction sequence.

Dataset	# sequence	# items	# actions	Avg_len	Sparsity
Beauty	22,363	12,101	198,502	8.53	99.93%
Toys	19,412	11,924	167,597	8.63	99.93%
Sports	25,598	18,357	296,337	8.32	99.95%
Steam	281,428	13,044	3,485,022	12.40	99.90%

4.1. Datasets

We, thereby, utilize four public datasets commonly used in the relevant literature for the overall performance comparison.

- *Amazon Beauty, Toys, Sports*[2] are collected from Amazon platform which contains numerous user–item interaction behaviors spanning from May 1996 to July 2014 (McAuley et al., 2015). Here, we choose the two most commonly used categories with different scales for model evaluation: *Amazon Beauty*, *Amazon Toys* and *Amazon Sports*.
- *Steam*[3] is obtained from an online video game website. The dataset collectes reviews from 334,730 users towards 13,047 games spanning from October 2010 to January 2018 and other external information (e.g., media score, users' play hours, games' prices and categories, publisher and developer information)

Following Kang and McAuley (2018), Li et al. (2020) and Wang, Wang et al. (2022), we regard the record of a rating or review as the user's implicit feedback, i.e., an interaction between item and user. Hence, for all the datasets, we filter inactive users and unpopular items whose interaction records are less than five. After that, for each user, we organize all of his or her historical interaction records chronologically as a sequence based on the item timestamp information associated with each interaction. Furthermore, we apply the leave-one-out protocol for all the datasets for performance evaluation. More concretely, given a sequence $\mathcal{S}_u = \{i_1, i_2, \dots, i_l\}$, we obtain the last interaction i_l for model testing, the penultimate interaction i_{l-1} for validation, and the remains $\{i_1, i_2, \dots, i_{l-2}\}$ for training. The statistics of the aforementioned four datasets are summarized in Table 1. We could observe that an interaction sequence has a shorter average length (8.53) on *Amazon Beauty* dataset, while *Steam* dataset has a longer length (12.40) instead.

4.2. Baselines and evaluation metrics

Baselines. We compare the Teddy against eight baselines from three groups: sequential neural network methods via using CNN, RNN and Transformer, denoising methods and multi-interest methods. The sequential neural network models are briefly introduced as below:

- **GRU4Rec**[4] employs a gated recurrent unit (GRU) to model the sequential patterns over the user's historical sequential interactions (Hidasi et al., 2015).
- **Caser**[5] utilizes a convolutional neural network (CNN) to model user's consecutive interaction behaviors over sub-sequences as well as the complicated feature level correlations for next item recommendation (Tang & Wang, 2018).
- **SASRec**[6] utilizes a uni-directional Transformer to model the complex patterns between the historical items (Kang & McAuley, 2018).
- **BERT4Rec**[7] assumes that the capability of uni-directional attention (e.g., SASRec) might be limited to model the user's complex sequential patterns. Hence, a bidirectional Transformer with cloze task is carried out for next-item prediction (Sun et al., 2019).

The denoising methods are briefly introduced as follows:

- **DSAN**[8] utilizes an adaptively sparse attention mechanism to diminish the negative impact of the irrelevant items towards the target candidate item embedding for sequence denoising and next item recommendation (Yuan et al., 2021).
- **FMLP**[9] integrates Fast Fourier Transform (FFT) with MLP layer to filter the noise representations in frequency domain Zhou et al. (2022). FMLP utilizes circular convolution to obtain a larger receptive field on the whole sequence than conventional convolution. It validates that the periodic patterns can be well exploited than using conventional convolution.

The multi-interest methods are briefly introduced as follows:

[2] https://cseweb.ucsd.edu/~jmcauley/datasets/amazon_v2/.
[3] https://steam.internet.byu.edu/.
[4] https://github.com/hidasib/GRU4Rec.
[5] https://github.com/graytowne/caser.
[6] https://github.com/kang205/SASRec.
[7] https://github.com/FeiSun/BERT4Rec.
[8] https://github.com/SamHaoYuan/DSANForAAAI2021.
[9] https://github.com/RUCAIBox/FMLP-Rec.

Table 2
Experimental results (%) on the four public datasets. We highlight the best results and the second-best results in boldface and underlined respectively.

Datasets	Metric	GRU4Rec	Caser	SASRec	BERT4Rec	ComiRec	TiMiRec	DSAN	FMLP	Teddy
Beauty	HR@5	1.0112	1.6188	3.2688	2.1326	2.0495	1.9044	2.0288	3.7192	**5.5778***
	HR@10	1.9370	2.8166	6.2648	3.7160	4.4545	3.3434	2.9965	6.6843	**7.4713***
	HR@20	3.8531	4.4048	8.9791	5.7922	7.6968	5.1674	4.3621	9.0901	**9.7754***
	NDCG@5	0.6084	0.9758	2.3989	1.3207	1.0503	1.2438	1.2914	1.8406	**4.0994***
	NDCG@10	0.9029	1.3602	3.2305	1.8291	1.8306	1.7044	1.6013	2.7991	**4.7100***
	NDCG@20	1.3804	1.7595	3.6536	2.3541	2.6451	2.1627	1.7483	4.3670	**5.2884***
Toys	HR@5	1.1009	0.9622	4.5333	1.9260	2.3026	1.1631	2.3853	3.5917	**5.9575***
	HR@10	1.8553	1.8317	6.5496	2.9312	4.2901	1.8169	0.9546	6.4665	**7.5024***
	HR@20	3.1827	2.9500	9.2263	4.5889	6.9357	2.7156	3.2311	**9.9358**	9.5637
	NDCG@5	0.6983	0.5707	3.0105	1.1630	1.1571	0.7051	1.1208	1.7948	**4.4524***
	NDCG@10	0.9396	0.8510	3.7533	1.4870	1.7953	0.9123	1.1321	2.7229	**4.9529***
	NDCG@20	1.2724	1.1293	4.3323	1.9038	2.4631	1.1374	1.5341	3.5935	**5.4700***
Sports	HR@5	0.7030	0.9724	1.7166	1.0943	1.0366	0.8294	0.7762	1.7700	**2.8218***
	HR@10	1.3725	1.7346	3.2472	1.9643	2.4815	1.5502	1.2387	3.1752	**3.9955***
	HR@20	2.6636	2.8860	5.3177	3.3960	4.5992	2.6782	1.9309	4.9165	**5.5420***
	NDCG@5	0.4086	0.5676	0.8948	0.6806	0.5085	0.4991	0.8434	0.9052	**2.0355***
	NDCG@10	0.6230	0.8112	1.3832	0.9595	0.9742	0.7308	1.2659	1.3572	**2.4136***
	NDCG@20	0.9475	1.1000	1.9035	1.3181	1.5058	1.0140	0.7714	1.7952	**2.8014***
Steam	HR@5	3.1210	3.4505	4.4205	4.9281	3.3003	5.4290	5.5943	4.9883	**5.7950***
	HR@10	5.6733	6.5863	7.7550	8.1075	6.6944	8.8356	9.1324	8.7471	**9.6035***
	HR@20	9.9166	11.3021	12.7055	12.8597	11.7563	13.8937	14.3108	14.3264	**15.0762***
	NDCG@5	1.8673	1.8551	2.7109	3.1168	1.6713	3.4937	3.5719	3.0543	**3.6897***
	NDCG@10	2.6859	2.8603	3.7808	4.1369	2.7608	4.5865	4.7072	4.2587	**4.9129***
	NDCG@20	3.7507	4.0441	5.0248	5.3303	4.0327	5.8567	6.0085	5.6611	**6.2884***

* Denotes a significant improvement of Teddy over the best results (student t-test with p-value < 0.05).

- **ComiRec**[10] is a popular multi-interest modeling method for recommendation (Cen et al., 2020). We choose the variant ComiRec-SA that extracts multiple interests with an attention mechanism for performance comparison since it achieves much better performance than the other variant (i.e., ComiRec-DR with dynamic routing).
- **TiMiRec**[11] is an up-to-date multi-interest modeling method for sequential recommendation (Wang, Wang et al., 2022). Compared with ComiRec, TiMiRec introduces the target item as a supervised signal to guide the multi-level interest generation in the training stage. Then, based on the adjusted multi-level interest distribution for the next-item recommendation.

Evaluation Metrics. We evaluate all methods with two common metrics: HR@K (Hit Rate) and NDCG@K (Normalized Discounted Cumulative Gain), where K is set to be $5, 10, 20$ respectively. HR@K calculates the proportion of the testing instances that the ground-truth is listed among the top K items recommended by each method. NDCG@K further evaluates the ground-truth ranking position in top-K list. The larger NDCG score indicates better ranking performance, i.e., the higher the ground-truth position in the recommendation list. We set NDCG@$K = 0$, if the rank exceeds K.

Considering sampling-based evaluation will incur inconsistent conclusions when the negative items is relatively limited (Chen, Zhao et al., 2022; Krichene & Rendle, 2022), we select all the items excluding the ground-truth item as candidates for performance evaluation. Following previous studies (Kang & McAuley, 2018; Sun et al., 2019; Zhou et al., 2022), we set the maximum sequence length N to be 50 for the four datasets. Additionally, we conduct a statistical significance test via performing the student *t-test*.

4.3. Experimental setup

For a fair comparison, we follow previous studies (Kang & McAuley, 2018; Sun et al., 2019) and selected Adam optimizer with initial learning rate to be 0.001 and also utilized an early stopping strategy (i.e., no improvement for ten consecutive epochs) to alleviate overfitting problem. All parameters are initialized by a Xavier normalization distribution. The batch size is set to be 1024. We fix both embedding dimension and also hidden-state size as 64. We set the dropout rate as 0.1 for all the datasets. For the adaptive masking mechanism module, the number of latest items c is under the set of {1,2,3}. For the TCN module, the number of TCN blocks L and filter size k are tuned under the set of {1,2,3,4}.

4.4. Overall comparison (RQ1)

Table 2 reports the final results of all methods over the four open datasets. We, hereby, can make the following observations:

[10] https://github.com/THUDM/ComiRec.

[11] https://github.com/THUwangcy/ReChorus/tree/CIKM22.

Table 3
Performance comparison (%) of different variants of Teddy. The best results are highlighted in boldface, and the second-best results are underlined.

Dataset	Ablation	HR@5	HR@10	HR@20	NDCG@5	NDCG@10	NDCG@20
Beauty	w/o TRD	0.3560	0.7297	1.5241	0.2252	0.3442	0.5439
	w/o DIV	5.2611	7.3209	9.7006	3.8285	4.4920	5.0898
	w Soft-Mask	5.3294	7.2541	9.7070	3.8566	4.4768	5.0927
	w Transformer	4.8681	7.1437	9.9154	3.4104	4.1462	4.8452
	w Self-Attention	5.4179	7.2767	9.6690	3.9509	4.5441	5.1478
	Teddy	**5.5778**	**7.4713**	**9.7754**	**4.0994**	**4.7100**	**5.2884**
Toys	w/o TRD	0.7134	1.4486	2.7508	0.4153	0.6513	0.9765
	w/o DIV	5.3879	7.0575	9.1479	4.0291	4.5681	5.0915
	w Soft-Mask	5.6046	7.2081	9.4215	4.2471	4.7653	5.3233
	w Transformer	4.9560	6.9503	9.4767	3.4315	4.0748	4.7607
	w Self-Attention	5.2970	6.9589	9.1287	3.9753	4.5110	5.0572
	Teddy	**5.9575**	**7.5024**	**9.5637**	**4.4524**	**4.9529**	**5.4700**

Firstly, as to the sequential neural methods, GRU4Rec and Caser achieve the worst results on most settings across the total nine methods. The main reason is that those two models can only exploit the patterns over the consecutive items (or features) in a sequence. This largely hinders the relevant feature learning over the latter. Since the built-in self-attention mechanism can capture complex yet high-order patterns, both SASRec and BERT4Rec obtain a large performance gain over both GRU4Rec and Caser. Although BERT4Rec is more powerful than SASRec by further including left-to-right self-attention mechanism, BERT4Rec is weaker than SASRec on *Amazon Beauty*, *Amazon Toys* and *Amazon Sports* datasets. On *Steam* dataset, however, BERT4Rec outperforms SASRec on all the evaluation metrics. Note that the length of an interaction sequence in *Steam* dataset is longer than the counterparts on the other three datasets (ref. Table 1). These results seem to be reasonable since the bidirectional Transformer is more effective for long-sequence modeling.

Secondly, multi-interest methods (i.e., ComiRec and TiMiRec) and denoising methods (i.e., DSAN, FMLP) do not exhibit substantial superiority in comparison to the conventional sequential models, particularly in SASRec. That is to say, these methods perform quite unstable across different datasets. More specifically, TimiRec achieves better performance on *Steam* datasets while ComiRec performs significantly better than TiMiRec on *Amazon Beauty*, *Amaozn Toys* and *Amazon Sports* datasets instead. Similarly, FMLP achieves superior performance than DSAN on *Amazon Beauty*, *Amazon Toys*, and *Amazon Sports* datasets, while the situation reverses on *Steam* dataset. These results suggest that the current denoising and multi-interest methods are just sub-optimal to capturing the user's preference respectively.

Lastly, our proposed Teddy is obviously superior to all the baselines in an overwhelming majority of settings across the four datasets. In particular, Teddy obtains up to 31.42%/47.90% and 5.23%/4.66% relative improvement against the best baseline for *Amazon Toys* and *Steam* datasets respectively in terms of HR/NDCG. The results reveal that disentangling major interest trend and interest diversity is more effective in capturing the user's preference to its maximum.

4.5. Ablation study (RQ2)

To further investigate the impact and effectiveness of each dedicated module made in Teddy, we conduct a series of ablation studies or replace the current component with possible variants. This set of experiments is performed on *Amazon Beauty* and *Amazon Toys* datasets. Similar observations are also made from other two datasets. In detail, a series of variants with regard to the adaptive masking module, the interest trend modeling module and the interest diversity modeling module are examined as follows:

- **w/o TRD**: detaches the interest trend modeling module and utilizes the interest diversity modeling module only for recommendation.
- **w/o DIV**: detaches the interest diversity modeling module from Teddy and utilizes the interest trend modeling module only for recommendation.
- **w Soft-Mask**: replaces adaptive (binary) masking mechanism with an attention mechanism, i.e., both $\mathcal{S}_u$ and $\mathcal{D}_u$ contain all the historical items but the embedding of each item x_j is weighted by r_j and $1 - r_j$, respectively (ref. Eq. (2)).
- **w Transformer**: replaces the stacked multi-blocks TCN module with the multi-blocks vanilla Transformer for the user's interest trend modeling. The other settings and components keep the same as Teddy for a fair comparison.
- **w Self-Attention**: replaces both two-layer MLP and max-pooling mechanism with self-attention mechanism utilized in Cen et al. (2020) for interest diversity modeling. The other settings and components keep the same as Teddy for a fair comparison.

Table 3 presents the performance of five variants as well as the full Teddy on the *Amazon Beauty* and *Amazon Toys* datasets. Firstly, we can find that interest trend modeling plays a dominating role in capturing the user's intentions. A tremendous performance decline is obtained after removing interest trend modeling from Teddy, e.g., the performance of variant *w/o TRD* is an order of magnitude less than the variants that include the interest trend modeling.

Also, by removing the diversity interest modeling module, *w/o DIV* experiences performance decline to some extent on all the settings. But this variant is still superior to many strong baselines including FMLP, SASRec, in most settings. This is consistent with

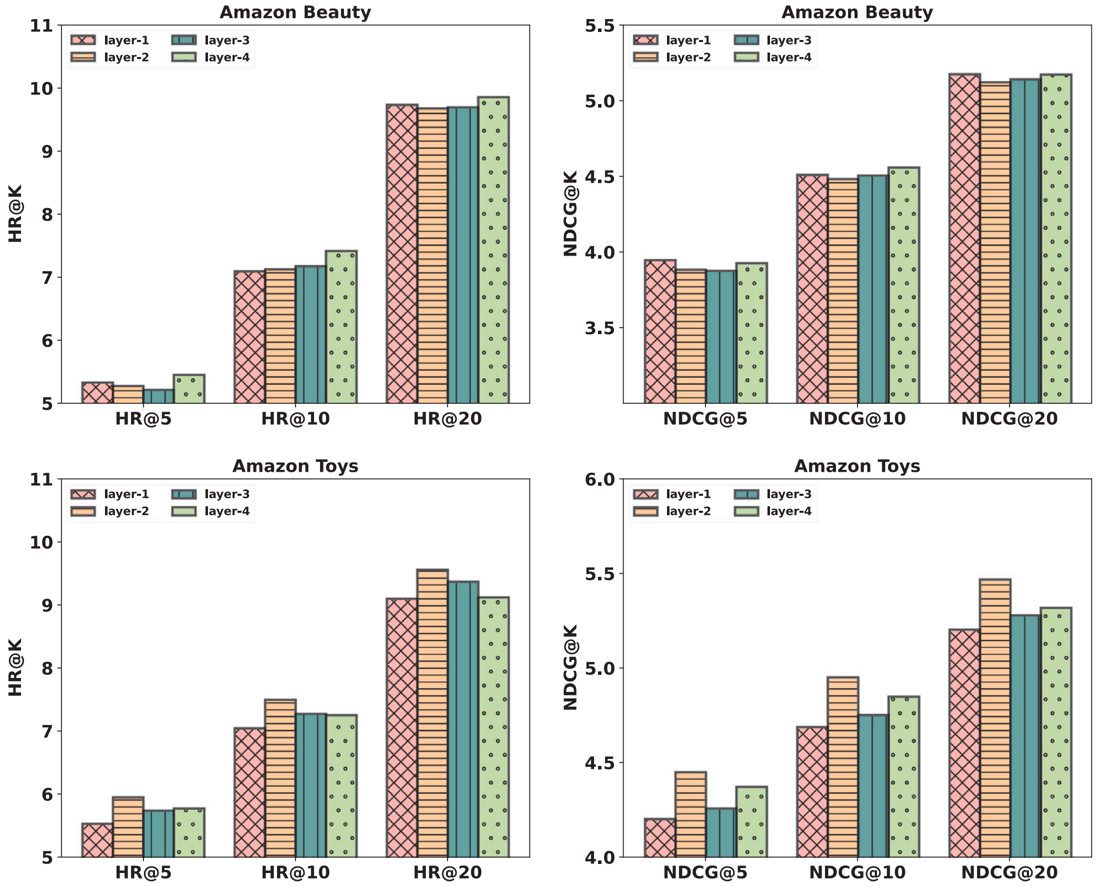

Fig. 3. Parameter sensitivity of the number of MLP layers.

the conclusion of existing works that denoising is an appropriate strategy on top of regular sequential models (Jin et al., 2022; Lin et al., 2022; Tong et al., 2021). In contrast, the disentanglement of user's interest trend and interest diversity proposed by Teddy is more effective and proper for understanding the user's preference.

Another interesting observation is that after replacing the stacked TCN with the vanilla Transformer (*w Transformer*) the performance experiences a slight decline. This suggests that the stacked TCN could be a powerful backbone for sequential recommendation. Moreover, when we replace the adaptive mask module and self-attention with *w Soft-Mask* and *w Self-Attention* module, the performance does not drop sharply. These results also confirm that each design choice contributes positively to Teddy.

4.6. Parameter analysis (RQ3)

Impact of MLP layers. In Section 3.5, a two layers MLP with max-pooling is carried out for interest diversity modeling. We, thereby, analyze the impact of MLP layers on the recommendation results. Fig. 3 depicts the performance of Teddy with different MLP layers. We could observe that the *HR* and *NDCG* exhibit minor fluctuations on *Amazon Beauty* dataset. In contrast, for *Amazon Toys* dataset, a moderate setting (i.e., 2) will be more effective and efficient.

Impact of TCN blocks. The number of TCN blocks is highly related to the sequence length. Fig. 4 plots the performance of Teddy across different blocks under the set of $\{1,2,3,4\}$. Here, we can see that large block numbers cannot generate better performance for *Amazon Beauty* and *Amazon Toys* datasets since the averaged sequence length is quite limited (i.e., 8.53 and 8.63, respectively). To recapitulate, one TCN block is sufficient for recommendation.

Impact of θ_m. Recall that hyper-parameter θ_m controls the discriminative capacity of disentangling interest trend and interest diversity. Here, we further analyze the impact of varying θ_m values (in the range of [−1,1]). When θ_m approaches -1, Teddy is equivalent to *w TCN*, while Teddy treats each historical item as scattered interests when θ_m approaches 1. Fig. 5 shows Teddy achieves more improvement within the positive θ_m on *Amazon Beauty* and *Amazon Toys* datasets, while a negative number (i.e., less than zero) deteriorates the performance for both two datasets. This is reasonable since a negative θ_m will introduce the items which opposite to the user's major interests and current intentions, hindering the model to achieve optimal results. However, it is also irrational to set an excessively large value (i.e., 0.8) as the user's whole interacted items will be dominated by the interest diversity. Consequently, a moderate value (i.e., $\theta_m = 0$) will be more proper for recommendation.

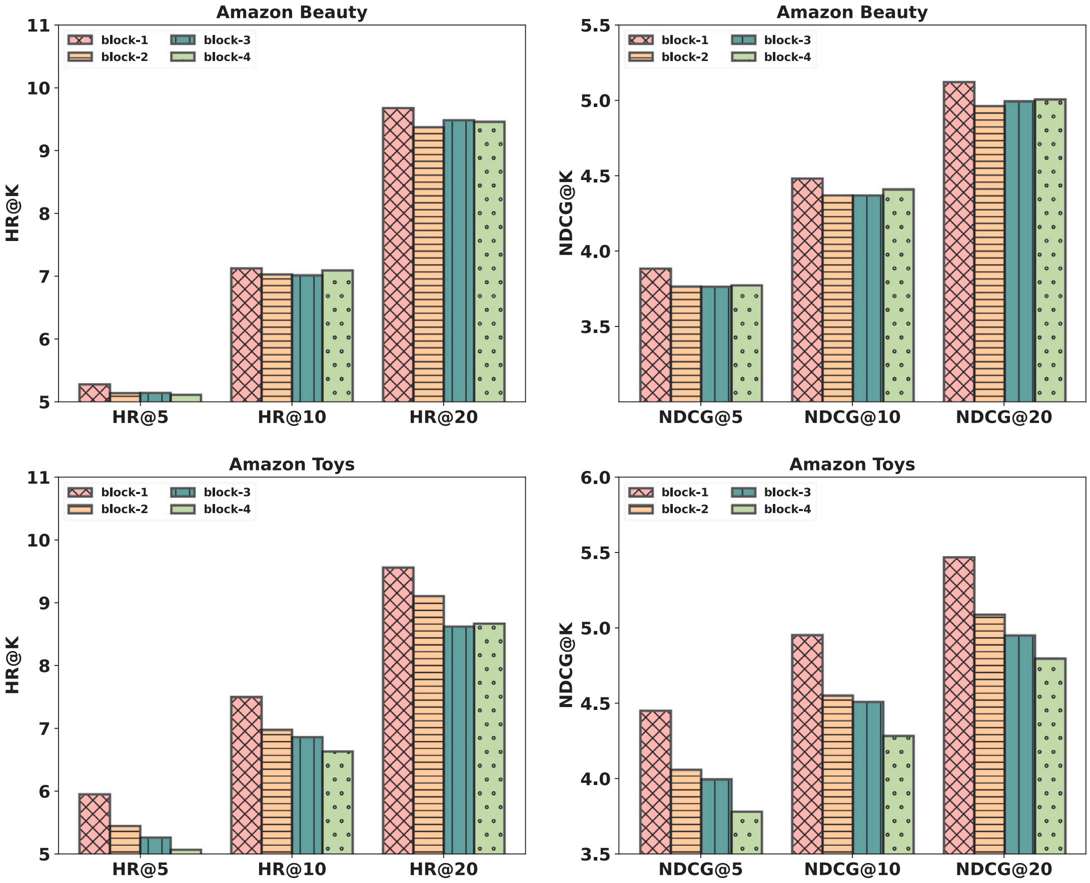

Fig. 4. Parameter sensitivity of the number of TCN blocks.

Impact of c. In formula (2), we designate the average pooling representation of the last c items as the supervised signal for item categorization (i.e., interest trend subsequence and interest diversity). Furthermore, we investigate the performance when only the cth last item (2nd and 3rd) is used as the indicator for item categorization. Note that, a small c (i.e., a few latest items are selected) will push the model to specialize more in user's current intentions and thus, more items toward the short-term interest will be classified to the interest trend. In contrast, a bigger c value (i.e., more items are contained) will augment the model's capacity for long-term interesting modeling, as both the user's current and previous intentions will be considered as the proxy for interest disentanglement. The results are presented in Table 4, we could find that for *Amazon Beauty* and *Amazon Baby* datasets, $c = 1$ is sufficient for interesting trend and interest diversity modeling and if we increase c, the model performance will decline gradually. This is because the sequence length of these two datasets is quite limited, as we set a large value, more irrelevant items will be contained for interest trend modeling, incurring the performance degeneration, which is also validated by the θ_m analysis (ref. Fig. 5). As for the results of 2nd and 3rd, we could draw a similar conclusion, i.e., TEDDY will attain the best performance when $c = 1$, and as c increases, the model's performance presents a small fluctuation for all the datasets in general. Thus, we believe $c = 1$ is the best setting for interest disentanglement.

Impact of α. As aforementioned described, we disentangle interest trend and interest diversity and model them in a holistic method. And the final prediction results are determined by these two modules (ref (10)). To instantiate the idea, we define a coupling factor α to arrange the allocation of proportions for each part. Therefore, we analyze the impact of α under the set of $\{0.2, 0.4, 0.6, 0.8\}$. It is noteworthy that when $\alpha = 0$, the model degrades to *w/o TRD* and if α closes to 1, the model is equivalent to $w/oDIV$. Fig. 6 illustrates the variation of α to the performance of TEDDY. We could observe that moderate values (i.e., $\alpha = 0.4$ or 0.6) are beneficial to the recommendation results, whereas, a small value will cause the model performance to have a sharp decrease, particularly on *Amazon Beauty* dataset. This phenomenon is also consistent with the experiment results of θ_m (Fig. 5), i.e., the model performance will diminish significantly when the information from user's interest diversity overwhelms the information from user's interest trend.

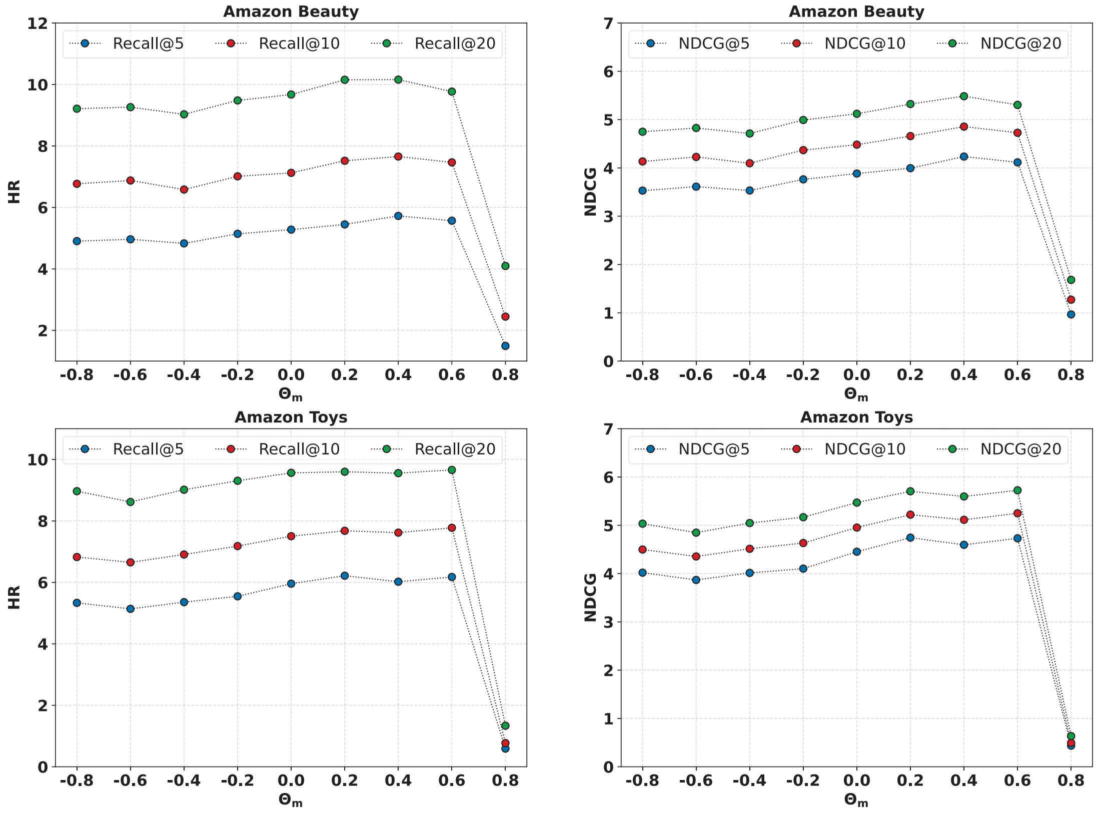

Fig. 5. Parameter sensitivity of the θ_m.

Table 4
Teddy performance over various c. 2nd and 3rd means the second and third of the last item respectively. The best results are highlighted in boldface, and the second-best results are underlined.

Dataset	c	HR@5	HR@10	HR@20	NDCG@5	NDCG@10	NDCG@20
Beauty	1	**5.5778**	**7.4713**	**9.7754**	**4.0994**	**4.7100**	**5.2884**
	2	5.2848	7.2073	9.6913	3.8847	4.5031	5.1286
	2nd	4.9637	6.7036	9.1695	3.6559	4.2156	4.8368
	3	5.0078	6.8550	9.3843	3.7135	4.3119	4.9468
	3rd	5.1599	6.8541	9.2096	3.7780	4.3235	4.9177
Toys	1	**5.9575**	**7.5024**	**9.5637**	**4.4524**	**4.9529**	**5.4700**
	2	5.4455	7.0779	9.2803	4.1207	4.6471	5.1988
	2nd	4.5858	5.9611	7.9634	3.5177	3.9586	4.4611
	3	5.1979	6.6664	8.6442	3.9078	4.3801	4.8779
	3rd	4.9271	6.4059	8.5327	3.6684	4.1475	4.6792

4.7. Further analysis (RQ4)

Proportion of Interest Trend. It is interesting to examine how many items are classified into the user's interest trend with respect to the sequence length. Consequently, we plot the proportion of items belonging to the interest trend against the whole historical sequence (i.e., $|I'|/|I|$) in Fig. 7 by using *Amazon Beauty* and *Amazon Toys* datasets. From our analysis, the following observations can be made: 1) Most instances have few interacted historical items available for modeling the user's preference. We can see that about 80% of historical sequences have less than 10 items; (2) The number of items classified into the user's interest trend generally surpasses the counterpart of the interest diversity. In addition, when the volume of historical items increases, this phenomenon becomes even more prominent. This also provides persuasive support that in each sequence, most items reveal one person's basic interests which will not shift dramatically over time, whereas the remaining items indicated the user's scattered interest diversity which is attributed to varying external factors; (3) Moreover, there is also a group of items that belong to the user's interest diversity. Consequently, these scattered interests are indispensable for understanding the user's preference to the maximum.

Case Study. At last, we dive deep into the testing instance to investigate whether the interacted historical items are highly correlated with each other in the interest trend part and diverse yet scattered in the interest diversity part respectively. To this end,

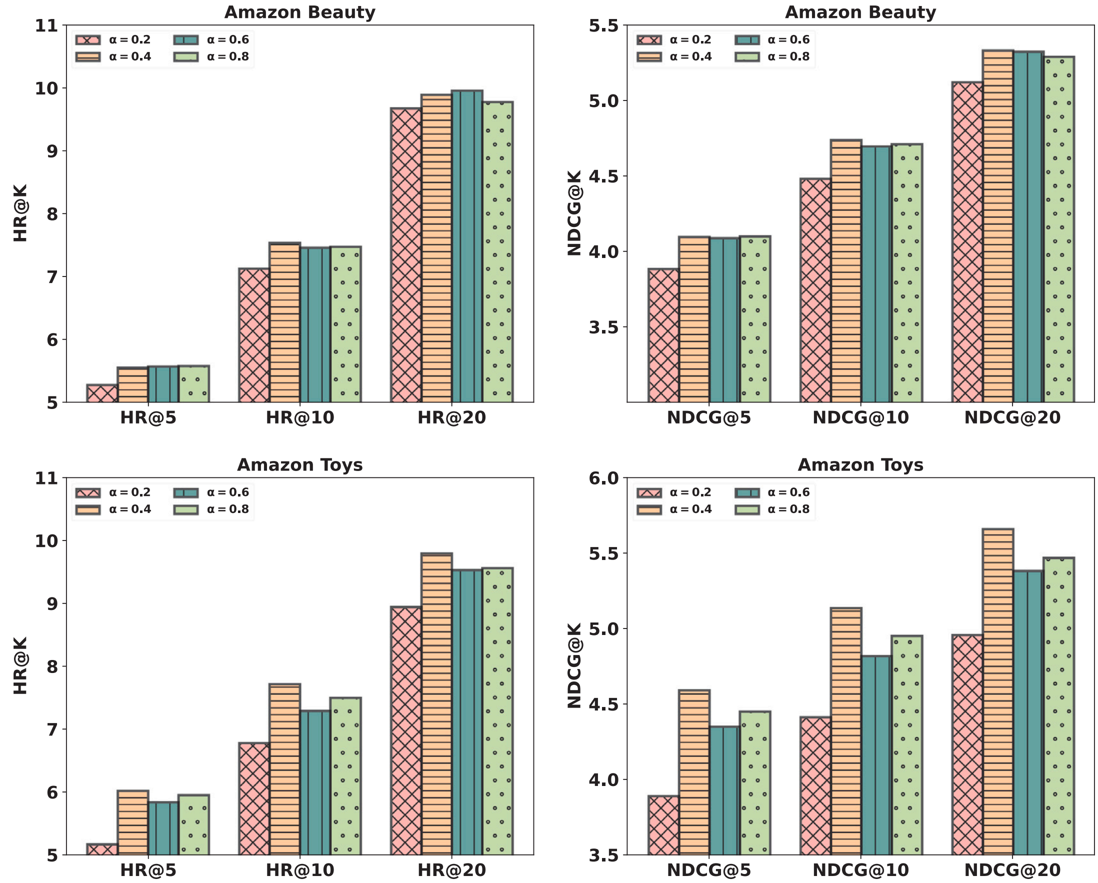

Fig. 6. Parameter sensitivity of the α.

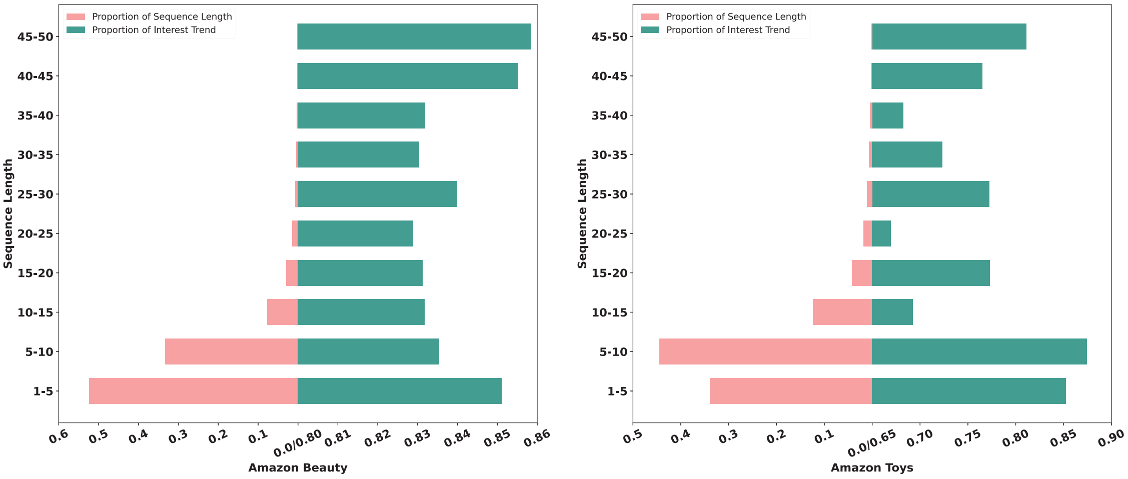

Fig. 7. The horizon bar of interest trend and sequence lengths distribution on *Amazon Beauty* and *Amazon Toys* datasets. The pink bar plot indicates the distribution of sequence length (The left side of the x-axis). The green bar plot illustrates the proportion distribution of items from interest trend to the whole sequence items (The right side of the x-axis.). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

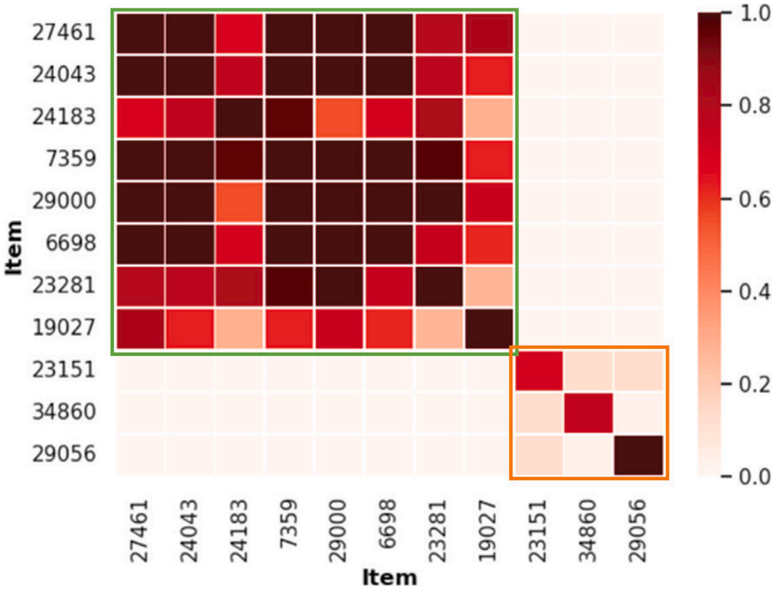

Fig. 8. The semantic correlations between items from the interest trend part (upper-left corner) and interest diversity part (lower-right corner) respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

we randomly pick one testing instance from the *Amazon Beauty* dataset and organize the sequence items into the above two parts according to the decision made by the adaptive masking mechanism (ref. Section 3.3). The pair-wise item embedding similarity for each part is calculated in terms of the inner product, as depicted in Fig. 8.

It is obvious that the items included for interest trend (circled by the green box) are highly correlated with each other, which also reveals the consistency of user's basic interest. On the other hand, the items of interest diversity (circled by the orange box) are highly irrelevant to each other instead, validating the existence of the user's scattered interest diversity.

5. Conclusion

In this paper, we attempt to model a user's interest trend and interest diversity separately but under a unified framework for sequential recommendation. To this end, we devise a simple yet effective model, named Teddy, which endeavors to disentangle a user's basic interest trend and scattered interest diversity and perform a customized strategy to model the user's interests from each part respectively. Extensive experiments demonstrate that our proposed Teddy achieves best performance against a series of state-of-the-art alternatives on four public datasets. Further analysis also offers obvious evidence towards the rationality that these two lines of perspectives are complementary to each other. In the future, we aim to introduce the knowledge graph to further enhance the discrimination between the interest trend and diversity.

CRediT authorship contribution statement

Zihao Li: Methodology, Writing, Investigation, Conceptualization. **Yunfan Xie:** Writing – review & editing. **Wei Emma Zhang:** Investigation, Writing – review & editing. **Pengfei Wang:** Investigation, Writing – review & editing. **Lixin Zou:** Supervision, Investigation, Writing – review & editing. **Fei Li:** Supervision, Investigation, Writing – review & editing. **Xiangyang Luo:** Investigation, Writing – review & editing. **Chenliang Li:** Supervision, Methodology, Investigation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62272349) and Young Top-notch Talent Cultivation Program of Hubei Province.

References

Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271.

Cen, Y., Zhang, J., Zou, X., Zhou, C., Yang, H., & Tang, J. (2020). Controllable multi-interest framework for recommendation. In *KDD* (pp. 2942–2951).

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

Chen, Y., Liu, Z., Li, J., McAuley, J., & Xiong, C. (2022). Intent contrastive learning for sequential recommendation. In *WWW* (pp. 2172–2182).

Chen, X., Xu, H., Zhang, Y., Tang, J., Cao, Y., Qin, Z., & Zha, H. (2018). Sequential recommendation with user memory networks. In *WSDM* (pp. 108–116).

Chen, G., Zhao, G., Zhu, L., Zhuo, Z., & Qian, X. (2022). Combining non-sampling and self-attention for sequential recommendation. *Information Processing & Management*, *59*(2), Article 102814.

Ding, Y., Ma, Y., Wong, W. K., & Chua, T.-S. (2021). Leveraging two types of global graph for sequential fashion recommendation. In *Proceedings of the 2021 international conference on multimedia retrieval* (pp. 73–81).

Du, H., Shi, H., Zhao, P., Wang, D., Sheng, V. S., Liu, Y., Liu, G., & Zhao, L. (2022). Contrastive learning with bidirectional transformers for sequential recommendation. arXiv preprint arXiv:2208.03895.

Fan, Z., Liu, Z., Wang, Y., Wang, A., Nazari, Z., Zheng, L., Peng, H., & Yu, P. S. (2022). Sequential recommendation via stochastic self-attention. In *WWW* (pp. 2036–2047).

Fan, Z., Liu, Z., Wang, S., Zheng, L., & Yu, P. S. (2021). Modeling sequences as distributions with uncertainty for sequential recommendation. In *CIKM* (pp. 3019–3023).

Fan, Z., Liu, Z., Zhang, J., Xiong, Y., Zheng, L., & Yu, P. S. (2021). Continuous-time sequential recommendation with temporal graph collaborative transformer. In *CIKM* (pp. 433–442).

Guo, L., Tang, L., Chen, T., Zhu, L., Nguyen, Q. V. H., & Yin, H. (2021). DA-GCN: a domain-aware attentive graph convolution network for shared-account cross-domain sequential recommendation. arXiv preprint arXiv:2105.03300.

Hidasi, B., & Karatzoglou, A. (2018). Recurrent neural networks with top-k gains for session-based recommendations. In *CIKM* (pp. 843–852).

Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2015). Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939.

Huang, J., Zhao, W. X., Dou, H., Wen, J.-R., & Chang, E. Y. (2018). Improving sequential recommendation with knowledge-enhanced memory networks. In *SIGIR* (pp. 505–514).

Jin, J., Chen, X., Zhang, W., Huang, J., Feng, Z., & Yu, Y. (2022). Learn over past, evolve for future: Search-based time-aware recommendation with sequential behavior data. In *Proceedings of the ACM web conference 2022* (pp. 2451–2461).

Kang, W.-C., & McAuley, J. (2018). Self-attentive sequential recommendation. In *ICDM* (pp. 197–206). IEEE.

Krichene, W., & Rendle, S. (2022). On sampled metrics for item recommendation. *Communications of the ACM*, *65*(7), 75–83.

Lea, C., Flynn, M. D., Vidal, R., Reiter, A., & Hager, G. D. (2017). Temporal convolutional networks for action segmentation and detection. In *CVPR* (pp. 156–165).

Li, A., Cheng, Z., Liu, F., Gao, Z., Guan, W., & Peng, Y. (2022). Disentangled graph neural networks for session-based recommendation. *IEEE Transactions on Knowledge and Data Engineering*.

Li, C., Liu, Z., Wu, M., Xu, Y., Zhao, H., Huang, P., Kang, G., Chen, Q., Li, W., & Lee, D. L. (2019). Multi-interest network with dynamic routing for recommendation at Tmall. In *CIKM* (pp. 2615–2623).

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

Lin, K., Wang, Z., Shen, S., Wang, Z., Chen, B., & Chen, X. (2022). Sequential recommendation with decomposed item feature routing. In *WWW* (pp. 2288–2297).

Liu, F., Chen, H., Cheng, Z., Liu, A., Nie, L., & Kankanhalli, M. (2022). Disentangled multimodal representation learning for recommendation. *IEEE Transactions on Multimedia*.

Liu, Q., Zeng, Y., Mokhosi, R., & Zhang, H. (2018). STAMP: short-term attention/memory priority model for session-based recommendation. In *KDD* (pp. 1831–1839).

Luo, A., Zhao, P., Liu, Y., Zhuang, F., Wang, D., Xu, J., Fang, J., & Sheng, V. S. (2020). Collaborative self-attention network for session-based recommendation. In *IJCAI* (pp. 2591–2597).

Ma, C., Kang, P., & Liu, X. (2019). Hierarchical gating networks for sequential recommendation. In *KDD* (pp. 825–833).

McAuley, J., Targett, C., Shi, Q., & Van Den Hengel, A. (2015). Image-based recommendations on styles and substitutes. In *SIGIR* (pp. 43–52).

Ni, S., Zhou, W., Wen, J., Hu, L., & Qiao, S. (2023). Enhancing sequential recommendation with contrastive Generative Adversarial Network. *Information Processing & Management*, *60*(3), Article 103331.

Pi, Q., Bian, W., Zhou, G., Zhu, X., & Gai, K. (2019). Practice on long sequential user behavior modeling for click-through rate prediction. In *KDD* (pp. 2671–2679).

Quadrana, M., Karatzoglou, A., Hidasi, B., & Cremonesi, P. (2017). Personalizing session-based recommendations with hierarchical recurrent neural networks. In *Proceedings of the eleventh ACM conference on recommender systems* (pp. 130–137).

Rendle, S., Freudenthaler, C., & Schmidt-Thieme, L. (2010). Factorizing personalized markov chains for next-basket recommendation. In *WWW* (pp. 811–820).

Shani, G., Heckerman, D., Brafman, R. I., & Boutilier, C. (2005). An MDP-based recommender system. *Journal of Machine Learning Research*, *6*(9).

Shao, W., Chen, X., Xia, L., Zhao, J., & Yin, D. (2022). Sequential recommendation with user evolving preference decomposition. arXiv preprint arXiv:2203.16942.

Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., & Jiang, P. (2019). BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *CIKM* (pp. 1441–1450).

Sun, Y., Wang, B., Sun, Z., & Yang, X. (2021). Does every data instance matter? Enhancing sequential recommendation by eliminating unreliable data. In *IJCAI* (pp. 1579–1585).

Tan, Q., Zhang, J., Liu, N., Huang, X., Yang, H., Zhou, J., & Hu, X. (2021). Dynamic memory based attention network for sequential recommendation. In *AAAI, vol. 35* (pp. 4384–4392).

Tang, J., & Wang, K. (2018). Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM* (pp. 565–573).

Taylor, W. L. (1953). "Cloze procedure": A new tool for measuring readability. *Journalism Quarterly*, *30*(4), 415–433.

Tian, Y., Chang, J., Niu, Y., Song, Y., & Li, C. (2022). When multi-level meets multi-interest: A multi-grained neural model for sequential recommendation. In *SIGIR* (pp. 1632–1641).

Tong, X., Wang, P., Li, C., Xia, L., & Niu, S. (2021). Pattern-enhanced contrastive policy learning network for sequential recommendation. In *IJCAI* (pp. 1593–1599).

Wang, S., Hu, L., Wang, Y., Cao, L., Sheng, Q. Z., & Orgun, M. (2019). Sequential recommender systems: challenges, progress and prospects. arXiv preprint arXiv:2001.04830.

Wang, C., Wang, Z., Liu, Y., Ge, Y., Ma, W., Zhang, M., Liu, Y., Feng, J., Deng, C., & Ma, S. (2022). Target interest distillation for multi-interest recommendation. In *Proceedings of the 31st ACM international conference on information & knowledge management* (pp. 2007–2016).

Wang, Y., Zhang, H., Liu, Z., Yang, L., & Yu, P. S. (2022). ContrastVAE: Contrastive variational AutoEncoder for sequential recommendation. In *CIKM* (pp. 2056–2066).

Wang, W., Zhang, W., Rao, J., Qiu, Z., Zhang, B., Lin, L., & Zha, H. (2020). Group-aware long-and short-term graph representation learning for sequential group recommendation. In *SIGIR* (pp. 1449–1458).

Wang, C., Zhu, Y., Liu, H., Ma, W., Zang, T., & Yu, J. (2021). Enhancing user interest modeling with knowledge-enriched itemsets for sequential recommendation. In *CIKM* (pp. 1889–1898).

Wu, L., Li, S., Hsieh, C.-J., & Sharpnack, J. (2020). SSE-PT: Sequential recommendation via personalized transformer. In *Fourteenth ACM conference on recommender systems* (pp. 328–337).

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

Xiao, T., Liang, S., & Meng, Z. (2019). Hierarchical neural variational model for personalized sequential recommendation. In *The world wide web conference* (pp. 3377–3383).

Yakhchi, S., Behehsti, A., Ghafari, S.-m., Razzak, I., Orgun, M., & Elahi, M. (2022). A convolutional attention network for unifying general and sequential recommenders. *Information Processing & Management*, *59*(1), Article 102755.

Ye, R., Zhang, Q., & Luo, H. (2020). Cross-session aware temporal convolutional network for session-based recommendation. In *2020 International conference on data mining workshops (ICDMW)* (pp. 220–226). IEEE.

Yuan, J., Song, Z., Sun, M., Wang, X., & Zhao, W. X. (2021). Dual sparse attention network for session-based recommendation. In *AAAI, vol. 35* (pp. 4635–4643).

Zhang, C., Du, Y., Zhao, X., Han, Q., Chen, R., & Li, L. (2022). Hierarchical item inconsistency signal learning for sequence denoising in sequential recommendation. In *CIKM* (pp. 2508–2518).

Zhang, S., Wu, Y., Che, T., Lin, Z., Memisevic, R., Salakhutdinov, R. R., & Bengio, Y. (2016). Architectural complexity measures of recurrent neural networks. *NeurIPS*, *29*.

Zhang, M., Wu, S., Yu, X., Liu, Q., & Wang, L. (2022). Dynamic graph neural networks for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*.

Zhang, S., Yang, L., Yao, D., Lu, Y., Feng, F., Zhao, Z., Chua, T.-s., & Wu, F. (2022). Re4: Learning to re-contrast, re-attend, re-construct for multi-interest recommendation. In *WWW* (pp. 2216–2226).

Zhou, K., Yu, H., Zhao, W. X., & Wen, J.-R. (2022). Filter-enhanced MLP is all you need for sequential recommendation. In *WWW* (pp. 2388–2399).

Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., Yan, Y., Jin, J., Li, H., & Gai, K. (2018). Deep interest network for click-through rate prediction. In *KDD* (pp. 1059–1068).

Zimdars, A., Chickering, D. M., & Meek, C. (2013). Using temporal data for making recommendations. arXiv preprint arXiv:1301.2320.