TME-PSR: Time-aware, Multi-interest, and Explanation Personalization for Sequential Recommendation

Anonymous ACL submission

Abstract

In this paper, we propose a sequential recommendation model that integrates **T**ime-aware personalization, **M**ulti-interest personalization, and **E**xplanation personalization for **P**ersonalized **S**equential **R**ecommendation (**TME-PSR**). That is, we consider differences across users in temporal rhythm preference, multiple fine-grained latent interests, and personalized semantic alignment between recommendations and explanations. Specifically, TME-PSR employs a dual-view gated time encoder to capture personalized temporal rhythms, a lightweight multihead Linear Recurrent Unit architecture to enable fine-grained sub-interest modeling with improved efficiency, and a dynamic dual-branch mutual information weighting mechanism to achieve personalized alignment between recommendations and explanations. Extensive experiments on real-world datasets demonstrate that our method consistently improves recommendation accuracy and explanation quality with lower computational cost.

1 Introduction

In recent years, personalized recommendation has attracted increasing attention as user expectations rise in recommendation systems. Personalized recommendation aims to select items (information or products) that best suit individual users' preferences based on their historical behaviors. Although recent advances in sequential recommendation (SR) models have significantly improved recommendation performance, most methods still fall short in delivering comprehensive personalization. In this paper, we aim to model three types of personalization simultaneously, including the following three perspectives.

• **Temporal rhythm preference**. Given a set of historical interactions of a specific user, the interaction time can reflect the personalized temporal rhythm preference of the user. For example, as fig. 1 shows, User 1 watches two romantic dramas on one day, and then, after a long gap of nearly a month, suddenly watches two action movies, which suggests a clear shift in temporal rhythm. In contrast, User 2 watches superhero movies regularly. To encode such long-short-term temporal rhythms across different users, some studies design specific network architectures (Yu et al., 2019; Zhang et al., 2021; Zheng et al., 2022; Dang et al., 2023; Li et al., 2020; Chen et al., 2022a; Liu et al., 2024c; Shi et al., 2023; Tran et al., 2023) to improve recommendation performance, such as using two attention branches to encode long-term and short-term temporal rhythms, respectively. In comparison, we propose a *flexible module to adaptively encode the differences in long-short-term temporal rhythms among different users, which can be plugged into existing models without changing the model's original architecture*. More crucially, the proposed module is not limited to recommendation tasks; it also encodes temporal rhythm preferences for the explanation task, separately for the same user.

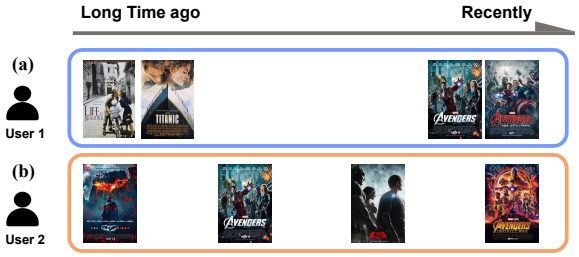

Figure 1: Examples of movie interaction sequences of two different users, which show different preferences for time and interests. User 1 liked romance movies a long time ago, but recently the interest has shifted to superhero movies. User 2 has always liked superhero movies, but also follows both Marvel and DC series as two sub-interests.

• **Multiple fine-grained latent interests**. A single sequence may contain multiple latent interests of a user, such as the interest in superhero movies and the interest in romance movies. More-

over, a user's latent interest can be further divided into **multiple fine-grained sub-interests**, such as the sub-interest in DC series movies and the sub-interest in Marvel series movies, as fig. 1(b) shows. Existing studies introduce an additional attention layer or an additional capsule network (Li et al., 2019; Chen et al., 2021; Cen et al., 2020; Zhang et al., 2022; Chai et al., 2022; Xie et al., 2023; Li et al., 2024) to learn multiple interests for different users. However, the time and space complexity of these methods increases with the number of interests. In comparison, we propose a method to *make the model automatically learn multiple fine-grained sub-interests with parallel network branches*, thereby disentangling them while improving efficiency and parameter economy.

• **Personalized semantic alignment between recommendations and explanations**. As explainability gains importance in recommendation systems, more users desire a textual rationale explaining why an item is recommended to them. Many studies, such as collaborative explanation (Li et al., 2021, 2023a; Yu et al., 2024), fuse recommendation features and explanation features to achieve semantic alignment between recommendations and explanations, thereby enabling explainable recommendation. However, these methods assume that the degree of semantic alignment between recommendations and explanations is uniform across users. Different users have different requirements for this alignment degree. For example, if a user often watched Marvel's superhero movies in the past, giving an explanation such as *"Iron Man 3, Robert Downey Jr. makes a powerful comeback"* can effectively entice the user to choose this movie; this user has high requirements for the semantic alignment degree between recommendations and explanations. In contrast, for users having a wide range of interests, this explanation might be less appealing, as they may not be familiar with *"Iron Man"* or *"Robert Downey Jr."* In this case, generating an explanation like *"An excellent action movie"* might be more likely to entice them to watch. These users have low requirements for the semantic alignment degree. Therefore, it is necessary to consider user differences in semantic alignment between recommendations and explanations, which is termed *explanation personalization*.

To this end, in this paper, we propose a sequential recommendation model to achieve the above three types of personalization. Specifically, as shown in fig. 2, the overall structure of our model is given as follows:

• To realize time-aware personalization, we propose a *dual-view gated time encoder module*. First, the module adaptively balances short-term and long-term behavior patterns based on each user's temporal rhythm preference. Second, for the same user, the module independently considers the temporal rhythm preference of the recommendation and the explanation task. The module is plug-and-play, requiring no changes to existing backbones.

• To realize multi-interest personalization, we propose to divide the representations of intermediate layers into multiple subspaces and track independent sub-interests in each subspace. Specifically, we extend the Linear Recurrent Unit (LRU) (Orvieto et al., 2023) to a *multihead LRU architecture* to disentangle multiple fine-grained latent sub-interests of a certain user while keeping lightweight.

• To realize explanation personalization, we design a *dynamic dual-branch mutual information weighting mechanism*, inspired by (Yu et al., 2024), to adaptively adjust the degree of semantic alignment between recommendations and explanations.

2 Related Work

Time-aware personalization. Time-aware methods incorporate temporal information to improve sequential recommendation. Early works utilized time matrix factorization (Chen et al., 2020; Ding and Li, 2005; Xiong et al., 2010) or attention mechanisms (Li et al., 2020; Zhang et al., 2021; Chen et al., 2022a) to encode time intervals. Later methods (Dang et al., 2023; Tran et al., 2023; Liu et al., 2024c) obtain more regular behavior sequences from temporal intervals. Recent studies further consider repeat consumption (Dai et al., 2024), temporal graph (Zhang et al., 2024), review-driven temporal signals (Shi et al., 2023), irregular temporal sparsity and item evolution (Fu et al., 2025), content-aware temporal attention (Liang et al., 2025), time-aware MoE (Zhang et al., 2025c), and periodic filtering (Baek et al., 2025). These methods confirm the importance of temporal signals, but many still require architecture-coupled designs.

Multi-interest personalization. To model users' multifaceted preferences, many methods learn multiple user vectors. Capsule/attention aggregators (Li et al., 2019; Cen et al., 2020) and their variants (Chen et al., 2021; Chai et al., 2022) extract several interests for matching/ranking. Recent methods

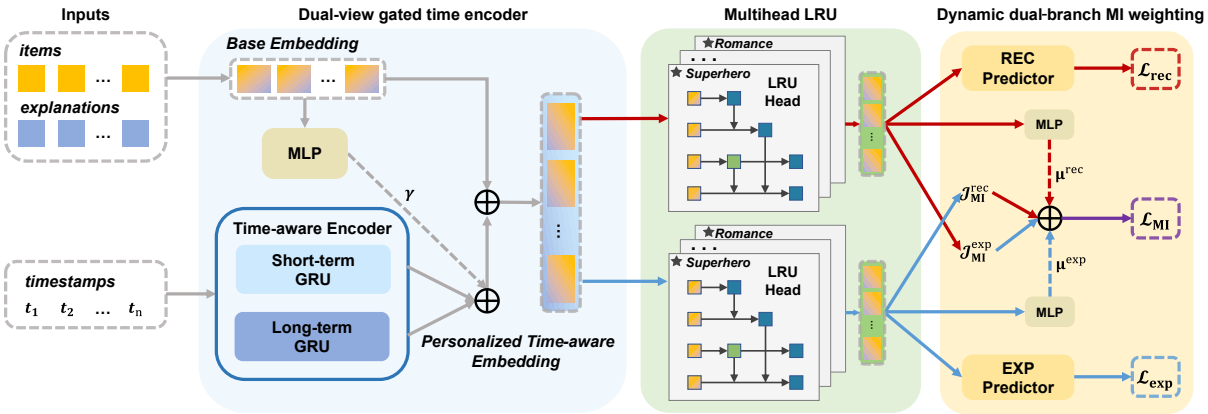

Figure 2: Overall architecture of our TME-PSR. At the input layer, we propose a dual-view gated time encoder. Specifically, we use a time-aware encoder to extract short-term behavior patterns and long-term behavior patterns from input timestamps. Then, we adaptively weight the two patterns based on different user's temporal rhythm preference, so as to extract personalized time-aware embeddings. At the encoding layer, we propose a multihead LRU architecture, each head captures a fine-grained sub-interest. At the prediction layer, we design a dynamic dual-branch MI weighting mechanism, so as to adaptively adjust the degree of semantic alignment between the recommendation branch and the explanation branch.

(Xia et al., 2022; Zhang et al., 2022; Xie et al., 2023; Du et al., 2024; Li et al., 2024) further improve interest decomposition and robustness. However, complexity often scales with interest count, leading to overlapping or redundant interests and higher computational costs.

Explanation personalization. Explainable SR provides textual rationales. Approaches utilize attention weights (Chen et al., 2023; Yang et al., 2021; Gholami et al., 2022; Li et al., 2023d), knowledge graphs (Cui et al., 2022; Geng et al., 2022a; Huang et al., 2019; Xian et al., 2019), or collaborative generation (Li et al., 2021, 2023a; Yu et al., 2024). However, existing collaborative explanation methods typically assume fixed recommendation–explanation alignment across users.

Broader advances in general sequential recommendation are discussed in App. A.

3 Method

Problem Formulation. Let $\mathcal{U}$ denote the set of users, $\mathcal{V}$ denote the set of items, and $\mathcal{E}$ denote the set of explanation labels.[1] For each user $u \in \mathcal{U}$, let $\mathcal{S}_u = [(v_1, e_1, t_1), (v_2, e_2, t_2), \ldots, (v_n, e_n, t_n)]$ denote the chronological interaction sequence of the user u, where at time step $i \in \{1, 2, ..., n\}$, the user action is a tuple of (v_i, e_i, t_i), $v_i \in \mathcal{V}$ is the item interacted with at time step i, $e_i \in \mathcal{E}$ is the associated explanation label, and $t_i \in \mathbb{R}^+$ is the interaction timestamp. We propose a dual-branch model TME-PSR, as fig. 2 shows, one branch predicts the next item v_{i+1} that user u is likely to interact with, while the other predicts the corresponding explanation label e_{i+1} for this item. The proposed TME-PSR model achieves time-aware personalization, multi-interest personalization, and explanation personalization, as follows.

[1]Following (Li et al., 2021), explanations are extracted from user reviews and clustered into representative phrases; each interaction is associated with an explanation label.

3.1 Time-Aware Personalization

To model user-specific temporal rhythm preference, we propose a *dual-view gated time encoder* to extract the personalized time-aware embedding from the timestamps of the user sequence $\mathcal{S}_u$. Specifically, we first encode two types of time intervals: the interval between adjacent items and the interval between each item and the first item in the sequence, thereby obtaining features containing short-term and long-term temporal information, respectively. Then, we employ a gating mechanism to adaptively fuse the short-term features and the long-term features, producing the personalized time-aware embedding of the user sequence S_u.

Base embedding construction. Let $\mathbf{M}_{\mathcal{V}} \in \mathbb{R}^{|\mathcal{V}| \times d}$ and $\mathbf{M}_{\mathcal{E}} \in \mathbb{R}^{|\mathcal{E}| \times d}$ denote the embedding matrices for all items $\mathcal{V}$ and all explanations $\mathcal{E}$, respectively, where d is the embedding size. Given a user sequence $\mathcal{S}_u$ with n interactions, we first apply a look-up operation from $\mathbf{M}_{\mathcal{V}}$ and $\mathbf{M}_{\mathcal{E}}$ to form the user's item embedding matrix $\mathbf{E}_{\mathcal{V}} \in \mathbb{R}^{n \times d}$ and explanation embedding matrix $\mathbf{E}_{\mathcal{E}} \in \mathbb{R}^{n \times d}$. We construct the recommendation base embed-

dings[2] $\mathbf{E}^{\text{rec}} \in \mathbb{R}^{n\times d}$ and the explanation base embeddings[2] $\mathbf{E}^{\text{exp}} \in \mathbb{R}^{n\times d}$ as follows, which will serve as a basis for constructing personalized time-aware embedding:

$$\mathbf{E}^{\text{rec}} = \alpha\cdot\mathbf{E}_{\mathcal{V}}+(1-\alpha)\cdot\mathbf{E}_{\mathcal{E}}, \quad \mathbf{E}^{\text{exp}} = \alpha\cdot\mathbf{E}_{\mathcal{E}}+(1-\alpha)\cdot\mathbf{E}_{\mathcal{V}}, \tag{1}$$

where $\alpha \in [0, 1]$ is a hyperparameter that controls the relative importance of each item embedding and its corresponding explanation embedding.

Dual-view gated time encoder. Let $\mathcal{T} = [t_1, t_2, ..., t_n] \in \mathbb{R}^n$ denote the timestamp sequence corresponding to the interaction sequence of the user u[2], we compute two types of time intervals to extract short-term and long-term temporal features:

(1) We compute the difference between adjacent timestamps t_i and t_{i-1} to obtain the *adjacent time interval* $\mathbf{\Delta}_i^{\text{adj}} = \log(1 + t_i - t_{i-1}) \in \mathbb{R}^+$, for $i = 2, 3, ..., n$, where the log operation is to ensure numerical smoothness; the constant 1 is added to avoid taking the log of zero.

(2) We compute the difference between each timestamp t_i and the first timestamp t_1 to obtain the *absolute time interval* $\mathbf{\Delta}_i^{\text{abs}} = \log(1+t_i-t_1) \in \mathbb{R}^+$, for $i = 2, 3, ..., n$.

Therefore, we construct the adjacent interval vector $\mathbf{\Delta}^{\text{adj}} = [0, \Delta_2^{\text{adj}}, ..., \Delta_n^{\text{adj}}] \in \mathbb{R}^n$ and the absolute interval vector $\mathbf{\Delta}^{\text{abs}} = [0, \Delta_2^{\text{abs}}, ..., \Delta_n^{\text{abs}}] \in \mathbb{R}^n$, where the constant 0 is prepended to each vector to align with the length n of the user sequence S_u to facilitate subsequent matrix operations.

To extract short- and long-term temporal features from $\mathbf{\Delta}^{\text{adj}}$ and $\mathbf{\Delta}^{\text{abs}}$, we employ two shared GRU (Chung et al., 2014) networks, $\text{GRU}_{\text{adj}}(\cdot) : \mathbb{R}^n \to \mathbb{R}^{n\times d}$ and $\text{GRU}_{\text{abs}}(\cdot) : \mathbb{R}^n \to \mathbb{R}^{n\times d}$, as time encoding networks:

$$\mathbf{H}^{\text{adj}} = \text{GRU}_{\text{adj}}(\mathbf{\Delta}^{\text{adj}}), \quad \mathbf{H}^{\text{abs}} = \text{GRU}_{\text{abs}}(\mathbf{\Delta}^{\text{abs}}), \tag{2}$$

where features[2] $\mathbf{H}^{\text{adj}} \in \mathbb{R}^{n\times d}$ represent the short-term temporal rhythm and features[2] $\mathbf{H}^{\text{abs}} \in \mathbb{R}^{n\times d}$ represent the long-term temporal rhythm.

To learn personalized time-aware embeddings, we propose to learn two gating parameters γ^{rec} and γ^{exp} for recommendation embeddings and explanation embeddings, respectively. These gates adaptively fuse the two features:

$$\tilde{\mathbf{E}}^{\text{rec}} = \gamma^{\text{rec}} \cdot \mathbf{H}^{\text{adj}} + (1 - \gamma^{\text{rec}}) \cdot \mathbf{H}^{\text{abs}}, \tag{3}$$

$$\tilde{\mathbf{E}}^{\text{exp}} = \gamma^{\text{exp}} \cdot \mathbf{H}^{\text{adj}} + (1 - \gamma^{\text{exp}}) \cdot \mathbf{H}^{\text{abs}}, \tag{4}$$

where the gating parameters γ^{rec} and γ^{exp} are learned as follows:

$$\gamma^{\text{rec}} = \sigma(\text{MLP}_{\text{rec}}(\bar{\mathbf{E}}^{\text{rec}})), \quad \gamma^{\text{exp}} = \sigma(\text{MLP}_{\text{exp}}(\bar{\mathbf{E}}^{\text{exp}})), \tag{5}$$

[2]The embeddings (e.g. $\mathbf{E}^{\text{rec}}$) and hidden states extracted in the Method section are all for a single user sequence S_u. The subscript u is omitted in these symbols for simplicity.

where $\text{MLP}_{\text{rec}}(\cdot) : \mathbb{R}^d \to \mathbb{R}^1$ and $\text{MLP}_{\text{exp}}(\cdot) : \mathbb{R}^d \to \mathbb{R}^1$ are two-layer perceptrons; $\bar{\mathbf{E}}^{\text{rec}} = \text{Mean1}(\mathbf{E}^{\text{rec}}) \in \mathbb{R}^{1\times d}$ and $\bar{\mathbf{E}}^{\text{exp}} = \text{Mean1}(\mathbf{E}^{\text{exp}}) \in \mathbb{R}^{1\times d}$; σ is sigmoid activation.

Finally, we obtain the personalized time-aware embeddings by combining the base embeddings[2] $\mathbf{E}^{\text{rec}}$ and $\mathbf{E}^{\text{exp}}$ with temporal features[2] $\tilde{\mathbf{E}}^{\text{rec}}$ and $\tilde{\mathbf{E}}^{\text{exp}}$:

$$\mathbf{E}^{\text{rec-time}} = \mathbf{E}^{\text{rec}} + \beta \cdot \tilde{\mathbf{E}}^{\text{rec}}, \quad \mathbf{E}^{\text{exp-time}} = \mathbf{E}^{\text{exp}} + \beta \cdot \tilde{\mathbf{E}}^{\text{exp}}, \tag{6}$$

where β is a scaling hyperparameter that controls the strength of temporal information.

3.2 Multi-interest Personalization

In this section, we further extract fine-grained sub-interests of users from the obtained personalized time-aware embeddings $\mathbf{E}^{\text{rec-time}}$ and $\mathbf{E}^{\text{exp-time}}$. To this end, we divide $\mathbf{E}^{\text{rec-time}}$ and $\mathbf{E}^{\text{exp-time}}$ into several sub-embeddings, and we extend LRU (Orvieto et al., 2023) to a multihead variant to extract sub-interests from these sub-embeddings. Specifically, given interaction (v_i, e_i, t_i) at time step i and its corresponding time-aware recommendation embedding $\mathbf{E}_i^{\text{rec-time}} \in \mathbb{R}^{1\times d}$ and explanation embedding $\mathbf{E}_i^{\text{exp-time}} \in \mathbb{R}^{1\times d}$, we divide the recommendation embedding $\mathbf{E}_i^{\text{rec-time}}$ into H sub-embeddings, $\mathbf{e}_i^{(\text{rec-time},1)}, ..., \mathbf{e}_i^{(\text{rec-time},H)} \in \mathbb{R}^{1\times \frac{d}{H}}$, and divide the explanation embedding $\mathbf{E}_i^{\text{exp-time}}$ into H sub-embeddings, $\mathbf{e}_i^{(\text{exp-time},1)}, ..., \mathbf{e}_i^{(\text{exp-time},H)} \in \mathbb{R}^{1\times \frac{d}{H}}$.

Then, we propose a multihead LRU architecture with H independent recurrent streams and we apply the architecture to both the recommendation branch and the explanation branch. Let $\mathbf{W}_h^{\text{rec}}, \mathbf{U}_h^{\text{rec}} \in \mathbb{R}^{\frac{d}{H}\times\frac{d}{H}}$ and $\mathbf{W}_h^{\text{exp}}, \mathbf{U}_h^{\text{exp}} \in \mathbb{R}^{\frac{d}{H}\times\frac{d}{H}}$ denote the learnable matrices for the h-th head in the recommendation branch and explanation branch, respectively. Each head learns to focus on a specific sub-interest. The hidden state for each head at time step i is computed as follows:

$$\mathcal{H}_i^{(\text{rec},h)} = \sum\nolimits_{k=1}^{i} \mathbf{e}_k^{(\text{rec-time},h)} (\mathbf{W}_h^{\text{rec}})^{i-k} \mathbf{U}_h^{\text{rec}} \in \mathbb{R}^{1\times\frac{d}{H}}, \tag{7}$$

$$\mathcal{H}_i^{(\text{exp},h)} = \sum\nolimits_{k=1}^{i} \mathbf{e}_k^{(\text{exp-time},h)} (\mathbf{W}_h^{\text{exp}})^{i-k} \mathbf{U}_h^{\text{exp}} \in \mathbb{R}^{1\times\frac{d}{H}}. \tag{8}$$

We stack the hidden states of all time steps to get the complete hidden state matrices $\mathcal{H}^{(\text{rec},h)}$ and $\mathcal{H}^{(\text{exp},h)}$ of the h-th head for the recommendation branch and explanation branch, respectively:

$$\mathcal{H}^{(\text{rec},h)} = \text{Stack}(\mathcal{H}_1^{(\text{rec},h)}, \mathcal{H}_2^{(\text{rec},h)}, ..., \mathcal{H}_n^{(\text{rec},h)}) \in \mathbb{R}^{n\times\frac{d}{H}}, \tag{9}$$

$$\mathcal{H}^{(\text{exp},h)} = \text{Stack}(\mathcal{H}_1^{(\text{exp},h)}, \mathcal{H}_2^{(\text{exp},h)}, ..., \mathcal{H}_n^{(\text{exp},h)}) \in \mathbb{R}^{n\times\frac{d}{H}}. \tag{10}$$

Finally, the personalized multi-interest representations[2] $\mathbf{Z}^{\text{rec}}$ for the recommendation branch and $\mathbf{Z}^{\text{exp}}$ for the explanation branch are obtained by concatenating all heads:

$$\mathbf{Z}^{\text{rec}} = \text{Concat}(\mathcal{H}^{(\text{rec},1)}, \mathcal{H}^{(\text{rec},2)}, ..., \mathcal{H}^{(\text{rec},H)}) \in \mathbb{R}^{n \times d}, \tag{11}$$

$$\mathbf{Z}^{\text{exp}} = \text{Concat}(\mathcal{H}^{(\text{exp},1)}, \mathcal{H}^{(\text{exp},2)}, ..., \mathcal{H}^{(\text{exp},H)}) \in \mathbb{R}^{n \times d}. \tag{12}$$

The proposed multihead LRU architecture not only captures multiple fine-grained sub-interests but also reduces the time and space complexity of the model. The time complexity of ideal parallel computing of all heads achieves $O(\log(n) \cdot (\frac{d}{H})^2)$. The space complexity achieves $O(\frac{d^2}{H})$. The per-step incremental inference complexity is $O(\frac{d^2}{H^2})$, independent of n. Please see App. F for the detailed complexity analysis. We demonstrate the advantages of the proposed multihead LRU on efficiency in fig. 5.

3.3 Explanation Personalization

To model personalized semantic alignment between recommendations and explanations, we extend (Yu et al., 2024) with a dynamic dual-branch mutual information (MI) weighting mechanism.

Dynamic dual-branch MI weighting mechanism. Given the hidden features[2] $\mathbf{Z}_i^{\text{rec}} \in \mathbb{R}^d$ and $\mathbf{Z}_i^{\text{exp}} \in \mathbb{R}^d$ at time step i, which contain time-aware personalized information and fine-grained multi-interest information, we use two learnable weights μ_i^{rec} and μ_i^{exp} to adaptively learn MI contributions for the recommendation branch and the explanation branch, respectively, as follows:

$$\mu_i^{\text{rec}} = \sigma(\text{MLP}_{\text{rec}}(\mathbf{Z}_i^{\text{rec}})), \mu_i^{\text{exp}} = \sigma(\text{MLP}_{\text{exp}}(\mathbf{Z}_i^{\text{exp}})), \tag{13}$$

where $\text{MLP}_{\text{rec}}(\cdot) : \mathbb{R}^d \to \mathbb{R}^1$ and $\text{MLP}_{\text{exp}}(\cdot) : \mathbb{R}^d \to \mathbb{R}^1$, with sigmoid activation $\sigma(\cdot)$. Let y_i^{rec} and y_i^{exp} be the target item and target explanation label at time step i. We define cross-branch alignment losses as follows:

$$\mathcal{J}_{\text{MI},i}^{\text{rec}} = -\log \frac{\exp(s_{i,y_i^{\text{rec}}}^{\text{rec}})}{\sum_{v \in \mathcal{V}} \exp(s_{i,v}^{\text{rec}})}, s_{i,v}^{\text{rec}} = (\mathbf{M}_{\mathcal{V}})_v \Lambda^{\text{rec}} (\mathbf{Z}_i^{\text{exp}})^\top, \tag{14}$$

$$\mathcal{J}_{\text{MI},i}^{\text{exp}} = -\log \frac{\exp(s_{i,y_i^{\text{exp}}}^{\text{exp}})}{\sum_{e \in \mathcal{E}} \exp(s_{i,e}^{\text{exp}})}, s_{i,e}^{\text{exp}} = (\mathbf{M}_{\mathcal{E}})_e \Lambda^{\text{exp}} (\mathbf{Z}_i^{\text{rec}})^\top, \tag{15}$$

where Λ^{rec} and Λ^{exp} are learnable projection matrices. The weighted MI objective is

$$\mathcal{J}_{\text{MI}} = \frac{1}{n} \sum_{i=1}^{n} \left[\mu_i^{\text{rec}} \cdot \mathcal{J}_{\text{MI},i}^{\text{rec}} + \mu_i^{\text{exp}} \cdot \mathcal{J}_{\text{MI},i}^{\text{exp}} \right]. \tag{16}$$

In this way, the term $\mathcal{J}_{\text{MI},i}^{\text{rec}}$ encourages each $\mathbf{Z}_i^{\text{exp}}$ to be closer to its paired recommendation embedding than to unpaired ones. The same applies to $\mathcal{J}_{\text{MI},i}^{\text{exp}}$ and each $\mathbf{Z}_i^{\text{rec}}$. The dynamic weights further allow each user sequence to determine how strongly the two branches should constrain each other.

3.4 Optimization

Prediction layer. Given the hidden features $\mathbf{Z}_i^{\text{rec}} \in \mathbb{R}^d$ from eq. (11) and $\mathbf{Z}_i^{\text{exp}} \in \mathbb{R}^d$ from eq. (12) at time step i, the prediction layer contains two functions $\text{Pred}(\mathbf{Z}_i^{\text{rec}}) : \mathbb{R}^d \to \mathbb{R}^{|\mathcal{V}|}$ and $\text{Pred}(\mathbf{Z}_i^{\text{exp}}) : \mathbb{R}^d \to \mathbb{R}^{|\mathcal{E}|}$. The function $\text{Pred}(\mathbf{Z}_i^{\text{rec}})$ produces the probabilities of the user u selecting the next item; the function $\text{Pred}(\mathbf{Z}_i^{\text{exp}})$ produces the degree of alignment between the item and all explanations:

$$\text{Pred}(\mathbf{Z}_i^{\text{rec}}) = \mathbf{M}_{\mathcal{V}}(\mathbf{Z}_i^{\text{rec}})^\top + \mathbf{b}^{\text{rec}}, \tag{17}$$

$$\text{Pred}(\mathbf{Z}_i^{\text{exp}}) = \mathbf{M}_{\mathcal{E}}(\mathbf{Z}_i^{\text{exp}})^\top + \mathbf{b}^{\text{exp}}, \tag{18}$$

where $\mathbf{b}^{\text{rec}} \in \mathbb{R}^{|\mathcal{V}|}$ and $\mathbf{b}^{\text{exp}} \in \mathbb{R}^{|\mathcal{E}|}$ are bias terms. Thus, the complete optimization objective is given as follows:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{exp}} + \mathcal{J}_{\text{MI}}. \tag{19}$$

The recommendation objective $\mathcal{L}_{\text{rec}}$ and the explanation objective $\mathcal{L}_{\text{exp}}$ are computed via cross-entropy loss based on the prediction scores $\text{Pred}(\mathbf{Z}^{\text{rec}})$ and $\text{Pred}(\mathbf{Z}^{\text{exp}})$.[3]

By optimizing the final objective $\mathcal{L}$, the proposed TME-PSR model achieves time-aware personalization, multi-interest personalization, and explanation personalization for sequential recommendation, thereby improving both recommendation and explanation performance.

4 Experiments

Models and datasets. We compare the proposed TME-PSR model with 4 types of SR approaches: (1) *General models*: GRU4Rec (Hidasi et al., 2015), SASRec (Kang and McAuley, 2018), FMLPRec (Zhou et al., 2022), CL4SRec (Xie et al., 2022), LRURec (Yue et al., 2024), and DiffuRec (Li et al., 2023e). (2) *Models with temporal modeling*: TLSAN (Zhang et al., 2021), MOJITO (Tran et al., 2023), and UniRec (Liu et al., 2024c). (3) *Models with multi-interest modeling*: ComiRec (Cen et al., 2020), Re4 (Zhang et al., 2022), and TEDDY (Li et al., 2024). (4) *Explainable models*: PITF (Rendle and Schmidt-Thieme, 2010) and BPER (Li et al., 2023a). Since the first three types of models lack explainability, we insert these 12 models as backbones into the explainable SR framework SCEMIM (Yu et al., 2024) (the experiments described in SCEMIM have demonstrated that it

[3]The formulations of $\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{\text{exp}}$ are provided in App. B.1.

Model	AM-Electronics				AM-Movies				Yelp			
	Recommendation		Explanation		Recommendation		Explanation		Recommendation		Explanation	
	R@10	N@10	R@10	N@10	R@10	N@10	R@10	N@10	R@10	N@10	R@10	N@10
PITF	0.0286	0.0139	0.0880	0.0547	0.0235	0.0101	0.0984	0.0679	0.0204	0.0094	0.0333	0.0185
BPER	0.0307	0.0154	0.0901	0.0553	0.0281	0.0131	0.1028	0.0687	0.0212	0.0106	0.0393	0.0214
SCEMIM-GRU4Rec	0.0290	0.0152	0.0948	0.0629	0.0432	0.0216	0.1151	0.0771	0.0452	0.0228	0.0417	0.0233
SCEMIM-SASRec	0.0369	0.0195	0.1011	0.0681	0.0541	0.0257	0.1229	0.0809	0.0592	0.0364	0.0419	0.0231
SCEMIM-FMLPRec	0.0388	0.0204	0.1032	0.0702	0.0558	0.0261	0.1192	0.0822	0.0639	0.0403	0.0408	0.0249
SCEMIM-LRURec	0.0327	0.0164	0.1087	0.0727	0.0549	0.0269	0.1171	0.0830	0.0650	0.0397	0.0422	0.0230
SCEMIM-CL4SRec	0.0367	0.0183	0.1090	0.0713	0.0515	0.0260	0.1213	0.0831	0.0536	0.0320	0.0424	0.0259
SCEMIM-DiffuRec	0.0402	0.0212	0.1186	0.0794	0.0544	0.0282	0.1352	0.0897	0.0624	0.0377	0.0498	0.0276
SCEMIM-ComiRec	0.0391	0.0185	0.1229	0.0725	0.0509	0.0262	0.1247	0.0817	0.0583	0.0358	0.0449	0.0273
SCEMIM-Re4	0.0390	0.0194	0.1170	0.0784	0.0487	0.0271	0.1207	0.0815	0.0608	0.0354	0.0455	0.0258
SCEMIM-TEDDY	0.0385	0.0206	0.1173	0.0735	0.0514	0.0275	0.1201	0.0854	0.0599	0.0367	0.0432	0.0260
SCEMIM-TLSAN	0.0383	0.0154	0.0957	0.0574	0.0457	0.0244	0.1012	0.0707	0.0492	0.0271	0.0375	0.0215
SCEMIM-MOJITO	0.0399	0.0201	0.1156	0.0690	0.0508	0.0280	0.1245	0.0840	0.0572	0.0354	0.0424	0.0262
SCEMIM-UniRec	0.0401	0.0211	0.1201	0.0790	0.0560	0.0301	0.1339	0.0924	0.0597	0.0386	0.0473	0.0270
TME-PSR	**0.0550**	**0.0291**	**0.1510**	**0.0979**	**0.0620**	**0.0339**	**0.1359**	**0.0931**	**0.0740**	**0.0455**	**0.0533**	**0.0309**
Improvement	36.82%	37.26%	22.86%	23.30%	10.71%	12.62%	0.52%	0.76%	13.87%	12.90%	7.03%	11.96%

Table 1: Recommendation and explanation performance. Best results are in bold and second-best results are underlined. The improvement row reports the relative gain of TME-PSR over the strongest baseline.

Strategy			AM-Electronics			
			Rec		Exp	
Time	Multi	Expl.	R@10	N@10	R@10	N@10
			0.0327	0.0164	0.1087	0.0727
✓			0.0402	0.0197	0.1371	0.0912
	✓		0.0361	0.0184	0.1312	0.0849
		✓	0.0324	0.0160	0.1252	0.0844
✓	✓		0.0498	0.0247	0.1455	0.0938
✓		✓	0.0450	0.0240	0.1423	0.0920
	✓	✓	0.0401	0.0234	0.1378	0.0915
✓	✓	✓	**0.0550**	**0.0291**	**0.1510**	**0.0979**

Table 2: Ablation of the three personalization strategies on AM-Electronics. Best results are in bold and second-best results are underlined.

can enable a model to acquire explainability while maintaining or even improving recommendation performance), thus obtaining 12 explainable SR models, denoted by SCEMIM-X (where X represents each model).

We conduct experiments on three widely used datasets for explainable SR, including the Amazon Movies dataset, the Amazon Electronics dataset, and the Yelp (Li et al., 2021) dataset. In these datasets, each user-item interaction is associated with a timestamp, an item ID, and an explanation ID. The explanations are extracted from user reviews and clustered into representative phrases through semantic hashing techniques (Li et al., 2021). We follow previous sequential recommendation work (Kang and McAuley, 2018; Yu et al., 2024) to divide each user's interaction history chronologically into three parts: the most recent interaction for testing, the second most recent for validation, and the remaining for training.[4] We evaluate both recommendation and explanation performance using two standard ranking metrics: Recall@K and NDCG@K, where K = 10. We refer to Recall@K as R@K and NDCG@K as N@K. We set $\alpha = 0.9$, $\beta = 0.1$, $d = 50$, and $H = 2$.[4]

Recommendation and explanation performance comparison. Tab. 1 shows the overall performance of all methods in recommendation and explanation across three datasets. As shown, our method consistently outperforms all baseline models across all evaluation metrics for both the recommendation task from 10.71% to 37.26% and the explanation task from 0.52% to 23.30%. These results verify the wide applicability and stability of the proposed model. A comparison with the baseline models without SCEMIM is provided in App. C. Results show that our method still provides a significant improvement.

Ablation Study. To further investigate the effectiveness of the three proposed personalized strategies, we evaluate every non-empty combination of the three personalization strategies on the AM-Electronics dataset, as tab 2 shows.[5] The main findings are as follows: (1) Time-aware personalization and multi-interest personalization are individually beneficial for both recommendation and explanation, while explanation personalization mainly im-

[4]Dataset statistics, hyperparameter analysis, and implementation details are provided in App. B.

[5]The ablation study on all three datasets is provided in App. D.

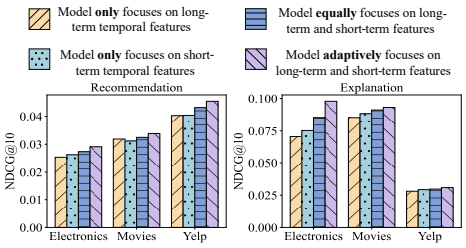

Figure 3: Comparison of four gating strategies. Results show that the model that adaptively focuses on both long-term and short-term features consistently achieves the best performance, supporting the effectiveness of the proposed method.

proves the explanation branch when used alone. (2) Pairwise combinations show clear synergy. For example, combining time-aware personalization and multi-interest personalization (row 5) outperforms any single component. (3) The full model is consistently the best. Enabling all three strategies attains the top performance on every metric. This confirms that the three types of personalization have significant mutual synergy.

Detailed analysis of the effectiveness of the time-aware personalization. In eq. (3) and eq. (4), we use gating parameters γ^{rec} and γ^{exp} to control the contribution of short-term and long-term temporal features to the recommendation and explanation time-aware embeddings. A larger γ means a greater contribution of short-term temporal features, while a smaller γ emphasizes long-term temporal features. To validate the dual-view gated time encoder, we compare four variants: using only long-term temporal features, using only short-term temporal features, equally combining both features, and adaptively combining them with our proposed gate. As shown in fig. 3, models that focus on both short-term and long-term temporal patterns outperform those that only focus on a single pattern, and dynamically balancing the two patterns further improves performance. We also found a linear relationship between the values of the two gating parameters and the average time interval of each user sequence; please see App. E for details.

Detailed analysis of the effectiveness of the multi-interest personalization. To further explore the effect of the head number H of the proposed multihead LRU architecture, we evaluate different head numbers H under varying embedding sizes $d \in \{60, 120, 240\}$, ensuring that H is a divisor of d. As fig. 4 shows, the optimal head number H^* consistently increases with performance on both the recommendation branch and the explanation branch: $H^* \approx 2, 6, 10$ for $d = 60, 120, 240$, and as the embedding size d increases, the overall performance improves. These findings indicate that higher dimensions of the representation space provide richer capabilities for interest disentanglement and require more fine-grained subspace decomposition. When H exceeds H^*, performance begins to degrade, likely due to each head being allocated too narrow a subspace. The third column of fig. 4 further shows that, when d remains fixed, the model parameters gradually decrease as H increases, which follows an inverse proportional relationship d^2/H.

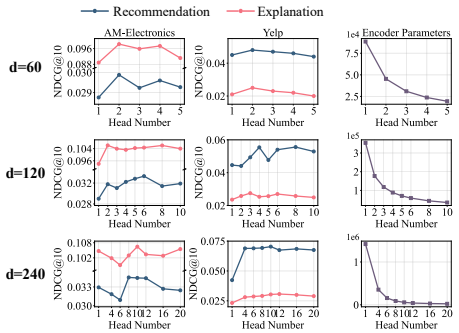

Figure 4: Left two columns show recommendation and explanation performance with different head numbers H under different embedding sizes d. Results show that the optimal head number increases with the increase of d. The third column shows the relationship between H and the encoder parameters. As H increases, the number of parameters decreases inversely.

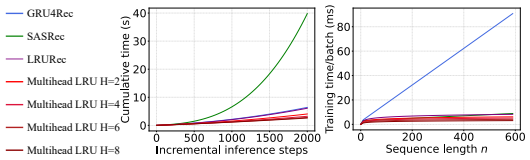

Figure 5: Inference and training efficiency of GRU4Rec, SASRec, LRURec and the proposed multihead LRU. Results show that the proposed method outperforms the other three methods, and its efficiency increases with the head number H.

We further explored incremental inference and training efficiency across GRU4Rec (Hidasi et al., 2015), SASRec (Kang and McAuley, 2018), LRURec (Yue et al., 2024), and the proposed multihead LRU ($H = 2, 4, 6, 8$) on embedding size $d = 240$, batch_size= 256. We perform only one batch of inference at each step, continuously expand the input length after each prediction, and test the training time required for a batch. As fig. 5 shows, the proposed method outperforms the compared three methods in terms of incremental inference and training efficiency, and the performance

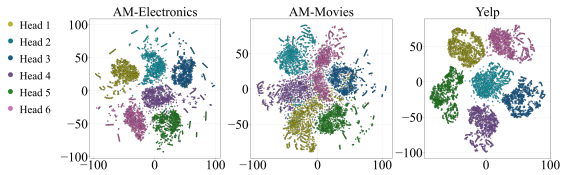

Figure 6: Visualization of multihead LRU output representation at $d = 120$, $H = 6$ with t-SNE.

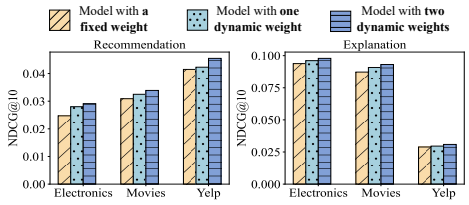

Figure 7: Comparison of three MI weighting strategies. Results show that the model with two dynamic weights consistently achieves the best performance, which supports the effectiveness of the proposed dynamic dual-branch MI weighting mechanism.

improvement is more significant with a larger number of heads. In summary, as d increases, increasing H appropriately can deliver higher accuracy, fewer parameters, and faster encoding.

To further explore whether the proposed multi-head LRU architecture truly captures distinct and fine-grained sub-interests, we performed a t-SNE (Maaten and Hinton, 2008) analysis on the output representations of each head under the conditions of $d = 120$ and $H = 6$. As fig. 6 shows, the six heads form well-separated clusters and cover complementary embedding regions.

Detailed analysis of the effectiveness of the explanation personalization. In eq. (16), two learnable weights μ^{rec} and μ^{exp} dynamically control the contribution of the recommendation MI loss and the explanation MI loss to the total MI loss. To validate this design, we compare three strategies: a fixed global weight 0.001, one learnable weight shared by the two MI losses, and the proposed dynamic dual-branch weights. As fig. 7 shows, learnable weights outperform a fixed weight, and dynamically learning two task-specific MI weights further improves performance, validating the effectiveness of the proposed dynamic dual-branch MI weighting mechanism.

Let $\mu^{\text{rec}}_{(u,i)}$ represent the MI weight of the recommendation task of the i-th interaction of user sequence S_u, and $\mu^{\text{exp}}_{(u,i)}$ represent the MI weight of the explanation task. We compute the average weights $\bar{\mu}^{\text{rec}}_u = \sum_{i=1}^{n} \mu^{\text{rec}}_{(u,i)}/n$ and $\bar{\mu}^{\text{exp}}_u = \sum_{i=1}^{n} \mu^{\text{exp}}_{(u,i)}/n$. Figure 8(a) reports the distributions of the averaged weights for all interaction sequences. Results show that all distributions are highly left-skewed and concentrated near zero, indicating that most sequences are assigned low semantic alignment strengths between recommendations and explanations in both tasks.[6] Meanwhile, the long tail indicates that a portion of sequences still require stronger semantic alignment.

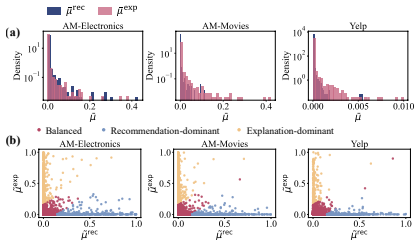

Figure 8: (a) Visualization of distributions of all interaction sequences' average dynamic MI weights $\bar{\mu}^{\text{rec}}$ and $\bar{\mu}^{\text{exp}}$. (b) K-means clustering based on $(\tilde{\mu}^{\text{rec}}, \tilde{\mu}^{\text{exp}})$.

To further investigate user-level differences in the degree of semantic alignment strength, we apply enhanced K-means clustering (De Amorim, 2016) on the following 2D point set, $(\tilde{\mu}^{\text{rec}}_u, \tilde{\mu}^{\text{exp}}_u)$, where $\tilde{\mu}^{\text{rec}}_u$ and $\tilde{\mu}^{\text{exp}}_u$ are normalized values of $\bar{\mu}^{\text{rec}}_u$ and $\bar{\mu}^{\text{exp}}_u$. Results show that user-level $\tilde{\mu}^{\text{rec}}_u$ and $\tilde{\mu}^{\text{exp}}_u$ exhibit three well-separated clusters shown in fig. 8(b): a balanced group near the diagonal, a recommendation-dominant group, and an explanation-dominant group. This proves that even for the same user, the recommendation task and the explanation task can have different requirements for semantic alignment, confirming the necessity of personalization.

5 Conclusion

This paper proposes TME-PSR, a sequential recommendation model that jointly addresses three types of personalization: time-aware, multi-interest, and explanation personalization. Our approach shows that personalization should be considered across multiple dimensions. Extensive experiments demonstrate that our model consistently improves recommendation accuracy, explanation quality, and efficiency across multiple datasets. For future work, we plan to extend personalization to cross-domain or multi-modal recommendation settings while maintaining explainability.

[6] Specific examples of the two types of user alignment requirements discussed in fig. 8(a) and the three types of users in fig. 8(b) are provided in App. G.

Limitations

TME-PSR uses explanation labels extracted from user reviews, and thus its explanation branch is influenced by the coverage and quality of the review-clustering process. Compared with state-of-the-art LLM-based explanation methods, this label-based setting may be less flexible in generating open-ended rationales. Nevertheless, by jointly modeling recommendation and explanation within the same sequential framework, TME-PSR remains lightweight, stable, and efficient. In addition, the current model mainly relies on item IDs, explanation labels, and timestamps. Incorporating multimodal content or cross-domain histories remains an important direction for future extension.

Ethics Statement

This work uses existing benchmark datasets and does not collect new user data or involve human-subject experiments. TME-PSR focuses on improving sequential recommendation and explanation modeling with item IDs, timestamps, and clustered explanation labels, and does not introduce mechanisms that directly infer or expose sensitive personal attributes. Therefore, we do not expect the proposed method to introduce additional ethical risks beyond standard recommendation-system concerns. In practical deployment, the model should still be used with appropriate privacy protection, fairness monitoring, and user-feedback mechanisms.

References

Ilwoong Baek, Mincheol Yoon, Seongmin Park, and Jongwuk Lee. 2025. Muffin: Mixture of user-adaptive frequency filtering for sequential recommendation. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 119–128.

Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD international conference*, pages 2942–2951.

Zheng Chai, Zhihong Chen, Chenliang Li, Rong Xiao, Houyi Li, Jiawei Wu, Jingxu Chen, and Haihong Tang. 2022. User-aware multi-interest learning for candidate matching in recommenders. In *Proceedings of the 45th SIGIR conference on research and development in information retrieval*, pages 1326–1335.

Gaode Chen, Xinghua Zhang, Yanyan Zhao, Cong Xue, and Ji Xiang. 2021. Exploring periodicity and interactivity in multi-interest framework for sequential recommendation. *arXiv preprint arXiv:2106.04415*.

Huiyuan Chen, Kaixiong Zhou, Zhimeng Jiang, Chin-Chia Michael Yeh, Xiaoting Li, Menghai Pan, Yan Zheng, Xia Hu, and Hao Yang. 2023. Probabilistic masked attention networks for explainable sequential recommendation. In *IJCAI*, pages 2068–2076.

Lihua Chen, Ning Yang, and Philip S Yu. 2022a. Time lag aware sequential recommendation. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pages 212–221.

Qi Chen, Guohui Li, Quan Zhou, Si Shi, and Deqing Zou. 2022b. Double attention convolutional neural network for sequential recommendation. *ACM Transactions on the Web*, 16(4):1–23.

Tong Chen, Hongzhi Yin, Quoc Viet Hung Nguyen, Wen-Chih Peng, Xue Li, and Xiaofang Zhou. 2020. Sequence-aware factorization machines for temporal predictive analytics. In *2020 IEEE 36th ICDE*, pages 1405–1416.

Jun-Young Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv: Neural and Evolutionary Computing*.

Zhihong Cui, Hongxu Chen, Lizhen Cui, Shijun Liu, Xueyan Liu, Guandong Xu, and Hongzhi Yin. 2022. Reinforced kgs reasoning for explainable sequential recommendation. *World Wide Web*, 25(2):631–654.

Sunhao Dai, Changle Qu, Sirui Chen, Xiao Zhang, and Jun Xu. 2024. Recode: Modeling repeat consumption with neural ode. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2599–2603.

Yizhou Dang, Enneng Yang, Guibing Guo, Linying Jiang, Xingwei Wang, Xiaoxiao Xu, Qinghui Sun, and Hong Liu. 2023. Uniform sequence better: Time interval aware data augmentation for sequential recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, pages 4225–4232.

Renato Cordeiro De Amorim. 2016. A survey on feature weighting based k-means algorithms. *Journal of Classification*, 33(2):210–242.

Gabriel de Souza Pereira Moreira, Sara Rabhi, Jeong Min Lee, Ronay Ak, and Even Oldridge. 2021. Transformers4rec: Bridging the gap between nlp and sequential/session-based recommendation. In *Proceedings of the 15th ACM conference on recommender systems*, pages 143–153.

Yi Ding and Xue Li. 2005. Time weight collaborative filtering. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 485–492.

Yujuan Ding, Yunshan Ma, Wai Keung Wong, and Tat-Seng Chua. 2021. Leveraging two types of global graph for sequential fashion recommendation. In *Proceedings of the 2021 international conference on multimedia retrieval*, pages 73–81.

Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential user-based recurrent neural network recommendations. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 152–160.

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

Yuanjun Feng, Stefan Feuerriegel, and Yash Raj Shrestha. 2025. Contextualizing recommendation explanations with llms: A user study. *arXiv preprint arXiv:2501.12152*.

Haoyan Fu, Zhida Qin, Shixiao Yang, Haoyao Zhang, Bin Lu, Shuang Li, Tianyu Huang, and John CS Lui. 2025. Time matters: Enhancing sequential recommendations with time-guided graph neural odes. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 637–648.

Shijie Geng, Zuohui Fu, Juntao Tan, Yingqiang Ge, Gerard De Melo, and Yongfeng Zhang. 2022a. Path language modeling over knowledge graphsfor explainable recommendation. In *Proceedings of the ACM web conference 2022*, pages 946–955.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022b. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM conference on recommender systems*, pages 299–315.

Ehsan Gholami, Mohammad Motamedi, and Ashwin Aravindakshan. 2022. Parsrec: Explainable personalized attention-fused recurrent sequential recommendation using session partial actions. In *Proceedings of the 28th ACM SIGKDD Conference*, pages 454–464.

Yingzhi He, Yan Sun, Junfei Tan, Yuxin Chen, Xiaoyu Kong, Chunxu Shen, Xiang Wang, An Zhang, and Tat-Seng Chua. 2026. Reasoning over semantic ids enhances generative recommendation. *arXiv preprint arXiv:2603.23183*.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.

Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*.

Yupeng Hou, Shanlei Mu, WayneXin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards universal sequence representation learning for recommender systems.

Xiaowen Huang, Quan Fang, Shengsheng Qian, Jitao Sang, Yan Li, and Changsheng Xu. 2019. Explainable interaction-driven user modeling over knowledge graph for sequential recommendation. In *proceedings of the 27th ACM international conference on multimedia*, pages 548–556.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

Jianchao Ji, Zelong Li, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Juntao Tan, and Yongfeng Zhang. 2024. Genrec: Large language model for generative recommendation. In *European Conference on Information Retrieval*, pages 494–502. Springer.

Yiheng Jiang, Yuanbo Xu, Yongjian Yang, Funing Yang, Pengyang Wang, Chaozhuo Li, Fuzhen Zhuang, and Hui Xiong. 2024. Trimlp: A foundational mlp-like architecture for sequential recommendation. *ACM Transactions on Information Systems*, 42(6):1–34.

Clark Mingxuan Ju, Liam Collins, Leonardo Neves, Bhuvesh Kumar, Louis Yufeng Wang, Tong Zhao, and Neil Shah. 2025. Generative recommendation with semantic ids: A practitioner's handbook. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 6420–6425.

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

Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at tmall. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 2615–2623.

Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th international conference on web search and data mining*, pages 322–330.

Lei Li, Yongfeng Zhang, and Li Chen. 2021. Extra: Explanation ranking datasets for explainable recommendation. In *Proceedings of the 44th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 2463–2469.

Lei Li, Yongfeng Zhang, and Li Chen. 2023a. On the relationship between explanation and recommendation: Learning to rank explanations for improved performance. *ACM Transactions on Intelligent Systems and Technology*, 14(2):1–24.

Lei Li, Yongfeng Zhang, and Li Chen. 2023b. Prompt distillation for efficient llm-based recommendation. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pages 1348–1357.

Xinhang Li, Chong Chen, Xiangyu Zhao, Yong Zhang, and Chunxiao Xing. 2023c. E4srec: An elegant effective efficient extensible solution of large language models for sequential recommendation. *arXiv preprint arXiv:2312.02443*.

Yicong Li, Hongxu Chen, Yile Li, Lin Li, Philip S Yu, and Guandong Xu. 2023d. Reinforcement learning based path exploration for sequential explainable recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(11):11801–11814.

Zihao Li, Aixin Sun, and Chenliang Li. 2023e. Diffurec: A diffusion model for sequential recommendation. *ACM Transactions on Information Systems*, 42(3):1–28.

Zihao Li, Yunfan Xie, Wei Emma Zhang, Pengfei Wang, Lixin Zou, Fei Li, Xiangyang Luo, and Chenliang Li. 2024. Disentangle interest trend and diversity for sequential recommendation. *Information Processing & Management*, 61(3):103619.

Siqi Liang, Yudi Zhang, and Yubo Wang. 2025. C-tlsan: Content-enhanced time-aware long-and short-term attention network for personalized recommendation. *arXiv preprint arXiv:2506.13021*.

Xiang Lin, Shuzi Niu, Yiqiao Wang, and Yucheng Li. 2018. K-plet recurrent neural networks for sequential recommendation. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1057–1060.

Chengkai Liu, Jianghao Lin, Jianling Wang, Hanzhou Liu, and James Caverlee. 2024a. Mamba4rec: Towards efficient sequential recommendation with selective state space models. *arXiv preprint arXiv:2403.03900*.

Qidong Liu, Xian Wu, Wanyu Wang, Yejing Wang, Yuanshao Zhu, Xiangyu Zhao, Feng Tian, and Yefeng Zheng. 2025. Llmemb: Large language model can be a good embedding generator for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 12183–12191.

Qidong Liu, Xian Wu, Xiangyu Zhao, Yejing Wang, Zijian Zhang, Feng Tian, and Yefeng Zheng. 2024b. Large language models enhanced sequential recommendation for long-tail user and item. *arXiv e-prints*, pages arXiv–2405.

Qidong Liu, Fan Yan, Xiangyu Zhao, Zhaocheng Du, Huifeng Guo, Ruiming Tang, and Feng Tian. 2023a. Diffusion augmentation for sequential recommendation. In *Proceedings of the 32nd ACM International conference on information and knowledge management*, pages 1576–1586.

Sijia Liu, Jiahao Liu, Hansu Gu, Dongsheng Li, Tun Lu, Peng Zhang, and Ning Gu. 2023b. Autoseqrec: Autoencoder for efficient sequential recommendation. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pages 1493–1502.

Yang Liu, Yitong Wang, and Chenyue Feng. 2024c. Unirec: A dual enhancement of uniformity and frequency in sequential recommendations. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 1483–1492.

Zihan Liu, Yupeng Hou, and Julian McAuley. 2024d. Multi-behavior generative recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 1575–1585.

Chao Long, Huanhuan Yuan, Junhua Fang, Xuefeng Xian, Guanfeng Liu, Victor S Sheng, and Pengpeng Zhao. 2024. Learning global and multi-granularity local representation with mlp for sequential recommendation. *ACM Transactions on Knowledge Discovery from Data*, 18(4):1–15.

Laurensvander Maaten and GeoffreyE. Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*.

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. 2023. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning*, pages 26670–26698. PMLR.

Bo Peng, Ziqi Chen, Srinivasan Parthasarathy, and Xia Ning. 2024. Modeling sequences as star graphs to address over-smoothing in self-attentive sequential recommendation. *ACM Transactions on Knowledge Discovery from Data*, 18(8):1–24.

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*, pages 548–556.

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*, pages 813–823.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, and 1 others. 2023. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36:10299–10315.

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

Steffen Rendle and Lars Schmidt-Thieme. 2010. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 81–90.

Xiaoyu Shi, Quanliang Liu, Yanan Bai, and Mingsheng Shang. 2023. Rtisr: a review-driven time interval-aware sequential recommendation method. *Journal of Big Data*, 10(1):32.

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

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*, pages 565–573.

Viet Anh Tran, Guillaume Salha-Galvan, Bruno Sguerra, and Romain Hennequin. 2023. Attention mixtures for time-aware sequential recommendation. In *Proceedings of the 46th SIGIR conference*, pages 1821–1826.

Jinpeng Wang, Ziyun Zeng, Yunxiao Wang, Yuting Wang, Xingyu Lu, Tianxiang Li, Jun Yuan, Rui Zhang, Hai-Tao Zheng, and Shu-Tao Xia. 2023. Missrec: Pre-training and transferring multi-modal interest-aware sequence representation for recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 6548–6557.

Ye Wang, Jiahao Xun, Minjie Hong, Jieming Zhu, Tao Jin, Wang Lin, Haoyuan Li, Linjun Li, Yan Xia, Zhou Zhao, and 1 others. 2024. Eager: Two-stream generative recommender with behavior-semantic collaboration. In *Proceedings of the 30th ACM SIGKDD Conference*, pages 3245–3254.

Zihao Wu, Xin Wang, Hong Chen, Kaidong Li, Yi Han, Lifeng Sun, and Wenwu Zhu. 2023. Diff4rec: Sequential recommendation with curriculum-scheduled diffusion augmentation. In *Proceedings of the 31st ACM international conference on multimedia*, pages 9329–9335.

Lianghao Xia, Chao Huang, Yong Xu, and Jian Pei. 2022. Multi-behavior sequential recommendation with temporal graph transformer. *IEEE Transactions on Knowledge and Data Engineering*, 35(6):6099–6112.

Yikun Xian, Zuohui Fu, Shan Muthukrishnan, Gerard De Melo, and Yongfeng Zhang. 2019. Reinforcement knowledge graph reasoning for explainable recommendation. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 285–294.

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

Yueqi Xie, Jingqi Gao, Peilin Zhou, Qichen Ye, Yining Hua, Jae Boum Kim, Fangzhao Wu, and Sunghun Kim. 2023. Rethinking multi-interest learning for candidate matching in recommender systems. In *Proceedings of the 17th ACM conference on recommender systems*, pages 283–293.

Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff Schneider, and Jaime G Carbonell. 2010. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proceedings of the 2010 SIAM international conference on data mining*, pages 211–222. SIAM.

Wujiang Xu, Qitian Wu, Zujie Liang, Jiaojiao Han, Xuying Ning, Yunxiao Shi, Wenfang Lin, and Yongfeng Zhang. 2025. Slmrec: Distilling large language models into small for sequential recommendation. In *Proceedings of the 13th International Conference on Learning Representations*, volume 2025, pages 35810–35831.

Zhengyi Yang, Jiancan Wu, Zhicai Wang, Xiang Wang, Yancheng Yuan, and Xiangnan He. 2023. Generate what you prefer: Reshaping sequential recommendation via guided diffusion. *Advances in Neural Information Processing Systems*, 36:24247–24261.

Zuoxi Yang, Shoubin Dong, and Jinlong Hu. 2021. Gfe: General knowledge enhanced framework for explainable sequential recommendation. *Knowledge-Based Systems*, 230:107375.

Yi Yu, Kazunari Sugiyama, and Adam Jatowt. 2024. Sequential recommendation with collaborative explanation via mutual information maximization. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1062–1072.

Zeping Yu, Jianxun Lian, Ahmad Mahmoody, Gongshen Liu, and Xing Xie. 2019. Adaptive user modeling with long and short-term preferences for personalized recommendation. In *IJCAI*, volume 7, pages 4213–4219.

Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. 2019. A simple convolutional generative network for next item recommendation. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 582–590.

Zhenrui Yue, Yueqi Wang, Zhankui He, Huimin Zeng, Julian McAuley, and Dong Wang. 2024. Linear recurrent units for sequential recommendation. In *Proceedings of the 17th ACM international conference on web search and data mining*, pages 930–938.

Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Michael He, Yinghai Lu, and Yu Shi. 2024. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. In *Proceedings of the 41st International Conference on Machine Learning*.

Jianqing Zhang, Dongjing Wang, and Dongjin Yu. 2021. Tlsan: Time-aware long-and short-term attention network for next-item recommendation. *Neurocomputing*, 441:179–191.

Jingsen Zhang, Zihang Tian, Xueyang Feng, and Xu Chen. 2025a. Enhancing recommendation explanations through user-centric refinement. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 8177–8191. Association for Computational Linguistics.

Qiang Zhang, Hanchao Yu, Ivan Ji, Chen Yuan, Yi Zhang, Chihuang Liu, Xiaolong Wang, Christopher E Lambert, Ren Chen, Chen Kovacs, and 1 others. 2025b. Efficient sequential recommendation for long term user interest via personalization. In *2025 IEEE International Conference on Data Mining (ICDM)*, pages 913–922. IEEE.

Shengyu Zhang, Lingxiao Yang, Dong Yao, Yujie Lu, Fuli Feng, Zhou Zhao, Tat-Seng Chua, and Fei Wu. 2022. Re4: Learning to re-contrast, re-attend, reconstruct for multi-interest recommendation. In *Proceedings of the ACM web conference 2022*, pages 2216–2226.

Shengzhe Zhang, Liyi Chen, Dazhong Shen, Chao Wang, and Hui Xiong. 2025c. Hierarchical time-aware mixture of experts for multi-modal sequential recommendation. In *Proceedings of the ACM on Web Conference 2025*, pages 3672–3682.

Shengzhe Zhang, Liyi Chen, Chao Wang, Shuangli Li, and Hui Xiong. 2024. Temporal graph contrastive learning for sequential recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 9359–9367.

Yifeng Zhang, Haohao Qu, Liangbo Ning, Wenqi Fan, and Qing Li. 2025d. Ssd4rec: A structured state space duality model for efficient sequential recommendation. *ACM Transactions on Information Systems*, 44(2):1–26.

Yu Zheng, Chen Gao, Jianxin Chang, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2022. Disentangling long and short-term interests for recommendation. In *Proceedings of the ACM web conference 2022*, pages 2256–2267.

Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Filter-enhanced mlp is all you need for sequential recommendation. In *Proceedings of the ACM web conference 2022*, pages 2388–2399.

A Additional Related Work

Sequential recommendation predicts a user's next interaction by modeling historical behavior. Early methods used Markov chains (Rendle et al., 2010), CNNs (Chen et al., 2022b; Tang and Wang, 2018; Yuan et al., 2019), and RNNs (Donkers et al., 2017; Hidasi et al., 2015; Hidasi and Karatzoglou, 2018; Lin et al., 2018); Transformer-based models (Kang and McAuley, 2018; Sun et al., 2019; de Souza Pereira Moreira et al., 2021) later improved long-range modeling and parallelism. Recent work broadens SR along several directions. Contrastive learning improves robustness through sequence augmentation and representation alignment (Xie et al., 2022; Qiu et al., 2022; Qin et al., 2024); diffusion models treat the target item as a denoising target conditioned on user history (Li et al., 2023e; Yang et al., 2023; Wu et al., 2023; Liu et al., 2023a); GNN-based SR (Liu et al., 2023b; Peng et al., 2024; Ding et al., 2021) models transition graphs; and all-MLP architectures (Zhou et al., 2022; Jiang et al., 2024; Long et al., 2024) trade attention for efficient filters.

Efficiency has become increasingly important. State-space and recurrent-style models such as Mamba4Rec and SSD4Rec adapt selective or structured state-space models to SR (Liu et al., 2024a; Zhang et al., 2025d), while HSTU scales generative recommenders to very long histories (Zhai et al., 2024). PerSRec compresses long histories into learnable personalized tokens for transformer-based recommenders (Zhang et al., 2025b). These backbones are relevant future baselines because they target the same efficiency–accuracy trade-off as our multihead LRU.

Another active line is semantic and language-based recommendation. Multi-modal SR methods pretrain on text/images and transfer across domains (Wang et al., 2023; Hou et al., 2022); generative retrieval decodes semantic IDs of items instead of ranking atomic IDs (Rajput et al., 2023; Liu et al., 2024d; Wang et al., 2024; Ju et al., 2025; He et al., 2026); and LLM-based recommenders use text-to-text task formulations (Geng et al., 2022b; Ji et al., 2024; Li et al., 2023b), semantic embeddings (Liu et al., 2025, 2024b; Li et al., 2023c), or distilled small language models (Xu et al., 2025). Recent

Dataset	Movies	Electronics	Yelp
# Users	22,248	18,129	90,742
# Items	15,458	10,435	46,372
# Explanations	31,451	23,541	119,874
# Actions	305,801	144,772	1,204,515
Avg. actions/user	13.75	7.95	13.27
Avg. actions/item	19.78	13.87	25.98
Avg. actions/expl.	9.72	6.15	10.05
User–item density	0.089%	0.076%	0.029%
User–expl. density	0.063%	0.034%	0.022%
User–item–expl. density	2.83e-6%	3.23e-6%	2.39e-7%

Table 3: Statistics of the preprocessed datasets.

LLM-based studies refine explanations along factuality, personalization, and sentiment dimensions (Zhang et al., 2025a), and examine how contextualized explanations affect user motivation and trust (Feng et al., 2025). These methods are complementary to TME-PSR: they enrich item/user semantics, whereas our focus is user-specific temporal, interest, and explanation-alignment personalization.

B Implementation Details

All experiments are conducted on a server equipped with 4×NVIDIA RTX 4090 GPUs (24GB each), an Intel Xeon Silver 4314 CPU @ 2.40GHz, and 256GB RAM. The software environment includes Ubuntu 18.04.4 LTS, Python 3.9.21, PyTorch 2.6.0+cu124, and CUDA 12.2.

B.1 Complete formulations of the recommendation objective function and the explanation objective function.

$$\mathcal{L}_{\text{rec}} = -\log \frac{\exp([\text{Pred}(\mathbf{Z}_i^{\text{rec}})]_{y_i^{\text{rec}}})}{\sum_{v\in\mathcal{V}} \exp([\text{Pred}(\mathbf{Z}_i^{\text{rec}})]_v)}, \quad (20)$$

$$\mathcal{L}_{\text{exp}} = -\log \frac{\exp([\text{Pred}(\mathbf{Z}_i^{\text{exp}})]_{y_i^{\text{exp}}})}{\sum_{e\in\mathcal{E}} \exp([\text{Pred}(\mathbf{Z}_i^{\text{exp}})]_e)}, \quad (21)$$

where y_i^{rec} is the target label at time step i in the recommendation branch; y_i^{exp} is the target label at time step i in the explanation branch.

B.2 Baselines

We compare our method with several representative sequential recommendation baselines:

• PITF (Rendle and Schmidt-Thieme, 2010): A pairwise-interaction tensor-factorization model using triadic (user, item, tag) embeddings to support explainable ranking.

• BPER (Li et al., 2023a): A Bayesian-personalized explanation-ranking model using joint BPR optimization to align item recommendations with textual rationales.

• GRU4Rec (Hidasi et al., 2015): A gated-recurrent-unit (Chung et al., 2014) model using pairwise ranking losses to learn within-session user–item transitions.

• SASRec (Kang and McAuley, 2018): A transformer-based sequential model leveraging self-attention to capture user-item transition patterns.

• FMLPRec (Zhou et al., 2022): A filter-based all-MLP model using learnable Fourier kernels to model sequential signals with low latency.

• LRURec (Yue et al., 2024): A linear-recurrence model using closed-form state transitions to approximate RNN behavior while enabling parallel computation.

• CL4SRec (Xie et al., 2022): A contrastive-learning SR model that augments sequences and jointly optimizes next-item prediction with self-supervised contrastive signals to learn stronger user representations.

• DiffuRec (Li et al., 2023e): A diffusion-based generative SR model that represents target items as distributions and denoises them from Gaussian noise conditioned on the user's history, improving recommendation under preference uncertainty.

• ComiRec (Cen et al., 2020): A controllable multi-interest framework that extracts multiple intent vectors (via capsule/attention variants) and matches candidates to the most relevant interest for each user.

• Re4 (Zhang et al., 2022): A multi-interest framework using re-contrast, re-attend and re-construct regularisers to maintain distinct intent vectors.

• TEDDY (Li et al., 2024): A disentangled SR method that separates interest trend and interest diversity (via adaptive masking and dedicated objectives) to improve robustness and personalization.

• TLSAN (Zhang et al., 2021): A dual-stream attention model using time-position embeddings for temporal drift.

• MOJITO (Tran et al., 2023): A time-aware Transformer that mixes temporal contexts with Gaussian-mixture attention, enabling better next-item prediction under varying temporal conditions.

• UniRec (Liu et al., 2024c): A uniformity–frequency enhanced SR model that exploits regularized time spacing and item-frequency cues via dual objectives to mitigate drift and sparsity.

In addition, SCEMIM (Yu et al., 2024) is an explainable sequential recommendation framework that enhances sequential recommendation with mutual information maximization between recommendation and explanation representations.

table 3 shows the statistics of the datasets.

B.3 Evaluation Metrics

Recall measures whether the ground-truth item (or explanation) appears in the top-k predictions. NDCG (Normalized Discounted Cumulative Gain) (Järvelin and Kekäläinen, 2002) further considers the position of correct predictions, assigning higher rewards to items ranked higher in the prediction list. Higher Recall and NDCG (Järvelin and Kekäläinen, 2002) values indicate better recommendation and explanation performance.

Recall@K. For a user u, let G_u be the set of relevant items in the test set and $\widehat{R}_u^{(K)}$ the top-K recommendations. The per-user recall is

$$\text{Recall@K}(u) = \frac{|\,\widehat{R}_u^{(K)} \cap G_u\,|}{|G_u|}. \tag{22}$$

Averaging over all test users $\mathcal{U}$ yields

$$\text{Recall@K} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|\,\widehat{R}_u^{(K)} \cap G_u\,|}{|G_u|}. \tag{23}$$

When $|G_u| = 1$ (next-item prediction), Recall@K reduces to Hit@K.

NDCG@K. Given the ranked list, let $r_{u,i}$ denote the relevance (1/0) of the item at position i for user u. The discounted cumulative gain is

$$\text{DCG}_u@K = \sum_{i=1}^{K} \frac{r_{u,i}}{\log_2(i+1)}. \tag{24}$$

The ideal DCG, obtained by perfect ranking, is

$$\text{IDCG}_u@K = \sum_{i=1}^{\min(K,|G_u|)} \frac{1}{\log_2(i+1)}. \tag{25}$$

We normalise DCG to

$$\text{NDCG}_u@K = \frac{\text{DCG}_u@K}{\text{IDCG}_u@K} \in [0,1], \tag{26}$$

and report the average over all users:

$$\text{NDCG@K} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \text{NDCG}_u@K. \tag{27}$$

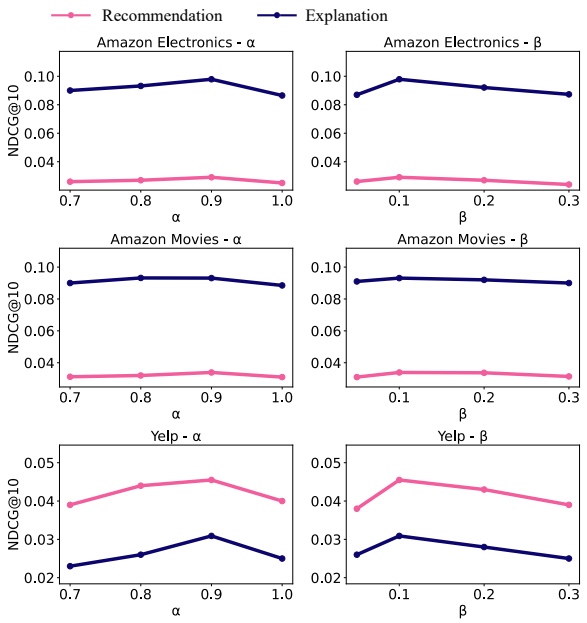

Figure 9: Impact of hyperparameters across three datasets. Results show that when set $\alpha = 0.9$ and set $\beta = 0.1$, the proposed model achieve the best performance.

B.4 Hyperparameter Sensitivity Analysis

Figure 9 plots NDCG@10 on the three datasets while sweeping the two scalar hyperparameters that control feature fusion in TME-PSR. The hyperparameter α determines the degree of fusion between the recommendation base embedding $\mathbf{E}_{\mathcal{V}}$ and the explanation base embedding $\mathbf{E}_{\mathcal{E}}$ features both in the recommendation branch and the explanation branch. The hyperparameter β controls the contribution of the temporal features $\tilde{\mathbf{E}}^{\text{rec}}$ and $\tilde{\mathbf{E}}^{\text{exp}}$ in the final personalized time-aware embeddings $\mathbf{E}^{\text{rec-time}}$ and $\mathbf{E}^{\text{exp-time}}$. When varying the value of one hyperparameter, we keep the other fixed. With β fixed to 0.1, we turn over α in $\{0.7, 0.8, 0.9, 1.0\}$. Increasing α from 0.7 to 0.9 consistently improves both recommendation performance and explanation performance. Then performances drops obviously when $\alpha = 1.0$, indicating the recommendation embedding and the explanation embedding are not fused. With α fixed to 0.9, we turn over β in $\{0.05, 0.1, 0.2, 0.3\}$. We observed that $\beta = 0.1$ yields the best results. Performance drop significantly when it goes beyond 0.1. On the other hand, when the β is too small, like 0.05, will cause the temporal feature to have too little impact to work.

Model	AM-Electronics		AM-Movies		Yelp	
	R@10	N@10	R@10	N@10	R@10	N@10
GRU4Rec	0.0259	0.0134	0.0431	0.0208	0.0451	0.0228
SASRec	0.0342	0.0182	0.0536	0.0256	0.0604	0.0364
FMLPRec	0.0357	0.0186	0.0558	0.0253	0.0620	0.0403
LRURec	0.0345	0.0159	0.0560	0.0259	0.0625	0.0397
CL4SRec	0.0371	0.0194	0.0523	0.0262	0.0510	0.0318
DiffuRec	0.0389	0.0201	0.0548	0.0273	0.0599	0.0325
ComiRec	0.0395	0.0198	0.0515	0.0271	0.0562	0.0333
Re4	0.0355	0.0185	0.0540	0.0265	0.0643	0.0400
TEDDY	0.0392	0.0208	0.0537	0.0279	0.0628	0.0411
TLSAN	0.0372	0.0191	0.0464	0.0245	0.0571	0.0323
MOJITO	0.0386	0.0198	0.0519	0.0302	0.0575	0.0364
UniRec	0.0403	0.0200	0.0557	0.0317	0.0578	0.0362
TME-PSR	**0.0550**	**0.0291**	**0.0620**	**0.0339**	**0.0740**	**0.0455**
Improvement	36.47%	39.90%	10.71%	6.94%	15.09%	10.71%

Table 4: Recommendation performance compared with non-SCEMIM baselines.

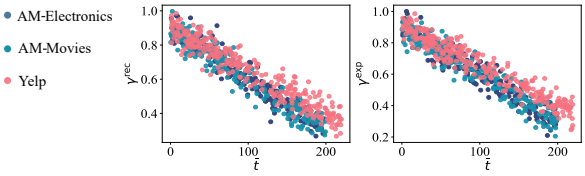

Figure 10: Relationship between the gating parameters γ^{rec} and the average time interval $\bar{t}$ (left) and relationship between the gating parameters γ^{exp} and $\bar{t}$ (right). Each point represents a single user sequence. Results show that as $\bar{t}$ increases, the two gating parameters decrease linearly.

C Complete comparisons with baseline models

We compare the proposed TME-PSR with the origin models without SCEMIM. As table 4 shows, the proposed method still achieved significant improvements in recommendation metrics across all three datasets.

D Complete ablation study

We evaluate every non-empty combination of the three personalization strategies on the three datasets, as table 5 shows. The main findings are entirely consistent with the ablation study described in sec 4. In summary, while the relative contribution of each component varies with the dataset characteristics, the complete TME-PSR configuration reliably delivers the strongest recommendation and explanation quality across all settings.

E Time-aware Personalization Details

We further explore the relationships between the two gating parameters γ^{rec} and γ^{exp} and the average time interval $\bar{t}$ of each user sequence. Specifically, for each user sequence S_u, we compute the average time interval $\bar{t} = \frac{1}{n-1}\sum_{i=2}^{n}(t_i - t_{i-1})$. Figure 10 visualizes 2D points $(\bar{t}, \gamma^{\text{rec}})$ and 2D points $(\bar{t}, \gamma^{\text{exp}})$ for different user sequences. Results show that as the average time interval $\bar{t}$ increases, the two gating parameters γ^{rec} and γ^{exp} decrease linearly. That is, sequences with short intervals tend to have larger γ^{rec} and γ^{exp}. This aligns with our design intuition: sequences with short intervals reflect recent, dense behaviors and thus benefit more from short-term temporal modeling, because large γ^{rec} and γ^{exp} mean a greater contribution of short-term temporal features to $\tilde{\mathbf{E}}^{\text{rec}}$ and $\tilde{\mathbf{E}}^{\text{exp}}$. In contrast, sequences with long intervals rely more on long-term temporal modeling.

F Complexity analysis of multihead LRU

We analyze a single-layer LRU for full-length forward with recursive parallelization. For one head with hidden size d, the time complexity is $O(\log(n) \cdot d^2)$ (Yue et al., 2024) for a sequence of length n. We ignore constant factors and linear-time terms due to normalization/residual wiring and do not account for additional FFN/projection costs here. Then, the time complexity of serially computing all heads is $O(H \cdot \log(n) \cdot (d/H)^2) = O(\log(n) \cdot \frac{d^2}{H})$. The time complexity of ideal parallel computing of all heads is $O(\log(n) \cdot (\frac{d}{H})^2)$. With per-head $\mathbf{W}_h, \mathbf{U}_h \in \mathbb{R}^{\frac{d}{H} \times \frac{d}{H}}$, the space complexity of per head is $O((\frac{d}{H})^2)$ and the total space complexity of the entire architecture is $O(\frac{d^2}{H})$, which is inversely proportional to the number of heads H. In comparison, typical transformer-based recommenders have a time complexity of $O(n^2 d + nd^2)$ and a space complexity of $O(d^2)$ and typical RNN-based recommenders have a time complexity of $O(nd^2)$ and a space complexity of $O(d^2)$.

G Examples of three representative user

Fig. 8(a) shows that all distributions are highly left-skewed and concentrated near zero, indicating that most sequences are assigned low semantic alignment strengths between recommendations and explanations in both the recommendation task and the explanation task. For example, User 1 in fig. 11(a) and User 3 in fig. 11(c) have low values of $\bar{\mu}^{\text{rec}}$ and $\bar{\mu}^{\text{exp}}$. Meanwhile, the long tail of the distribution ensures a portion of sequences still receive strong semantic alignment signals when needed. For example, User 2 in fig. 11(b) has high values of $\bar{\mu}^{\text{rec}}$

Strategy			AM-Electronics				AM-Movies				Yelp			
Time	Multi	Expl.	Rec		Exp		Rec		Exp		Rec		Exp	
			R@10	N@10	R@10	N@10	R@10	N@10	R@10	N@10	R@10	N@10	R@10	N@10
			0.0327	0.0164	0.1087	0.0727	0.0549	0.0269	0.1171	0.0830	0.0650	0.0397	0.0422	0.0230
✓			0.0402	0.0197	0.1371	0.0912	0.0563	0.0266	0.1287	0.0841	0.0672	0.0410	0.0454	0.0250
	✓		0.0361	0.0184	0.1312	0.0849	0.0550	0.0274	0.1300	0.0897	0.0652	0.0399	0.0428	0.0234
		✓	0.0324	0.0160	0.1252	0.0844	0.0567	0.0272	0.1242	0.0868	0.0654	0.0399	0.0437	0.0238
✓	✓		0.0498	0.0247	0.1455	0.0938	0.0583	0.0295	0.1294	0.0872	0.0699	0.0425	0.0476	0.0262
✓		✓	0.0450	0.0240	0.1423	0.0920	0.0592	0.0286	0.1274	0.0898	0.0689	0.0409	0.0488	0.0275
	✓	✓	0.0401	0.0234	0.1378	0.0915	0.0587	0.0300	0.1288	0.0913	0.0685	0.0412	0.0469	0.0259
✓	✓	✓	**0.0550**	**0.0291**	**0.1510**	**0.0979**	**0.0620**	**0.0339**	**0.1359**	**0.0931**	**0.0740**	**0.0455**	**0.0533**	**0.0309**

Table 5: Full ablation results on all datasets. Best results are in bold and second-best results are underlined.

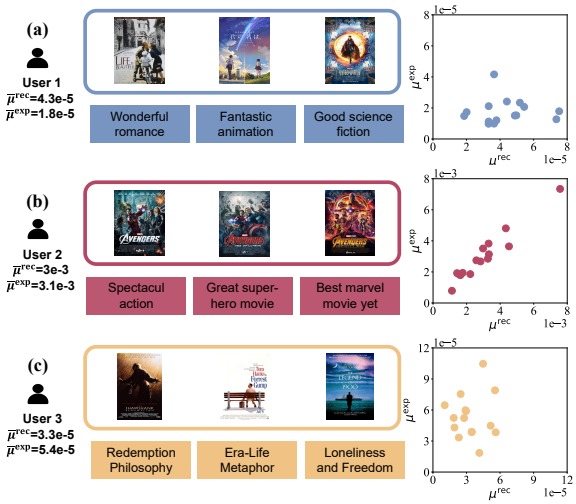

Figure 11: Examples of three representative user types, correspond to the three clusters in Figure 7(b).

and $\bar{\mu}^{\text{exp}}$.

Fig. 8(b) shows that user-level $\tilde{\mu}_u^{\text{rec}}$ and $\tilde{\mu}_u^{\text{exp}}$ exhibit three well-separated clusters: (1) The user shown in fig. 11(b) belongs to the balanced group, this type of users has similar semantic alignment requirements in the recommendation task and the explanation task. (2) The user shown in fig. 11(a) belongs to the recommendation-dominant group, this type of users has higher semantic alignment requirements in the recommendation task. (3) The user shown in fig. 11(c) belongs to the explanation-dominant group, this type of users has higher semantic alignment requirements in the explanation task.